



PCT/FR 00 / 0 1 9 5 0

REC'D 16 AUG 2000

WIPO

PCT

BREVET D'INVENTION

FR 00/1950

CERTIFICAT D'UTILITÉ - CERTIFICAT D'ADDITION

COPIE OFFICIELLE

Le Directeur général de l'Institut national de la propriété industrielle certifie que le document ci-annexé est la copie certifiée conforme d'une demande de titre de propriété industrielle déposée à l'Institut.

Fait à Paris, le 27 JUIN 2000

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)

Pour le Directeur général de l'Institut
national de la propriété industrielle
Le Chef du Département des brevets

Martine PLANCHE

INSTITUT
NATIONAL DE
LA PROPRIÉTÉ
INDUSTRIELLE

SIEGE
26 bis, rue de Saint Petersburg
75800 PARIS Cédex 08
Téléphone : 01 53 04 53 04
Télécopie : 01 42 93 59 30

THIS PAGE BLANK (USPTO)

REQUÊTE EN DÉLIVRANCE

26 bis, rue de Saint Pétersbourg
75800 Paris Cedex 08
Téléphone : 01 53 04 53 04 Télécopie : 01 42 93 59 30

Confirmation d'un dépôt par télécopie ☐

Cet imprimé est à remplir à l'encre noire en lettres capitales

Réservé à l'INPI

DATE DE REMISE DES PIÈCES **9 JUIL 1999**
N° D'ENREGISTREMENT NATIONAL **9908945**
DÉPARTEMENT DE DÉPÔT **75 INPI PARIS**
DATE DE DÉPÔT **09 JUIL 1999**

1 NOM ET ADRESSE DU DEMANDEUR OU DU MANDATAIRE
À QUI LA CORRESPONDANCE DOIT ÊTRE ADRESSÉE

CABINET PLASSERAUD
84, RUE D'AMSTERDAM
F-75440 PARIS CEDEX 09

2 DEMANDE Nature du titre de propriété industrielle

☒ brevet d'invention ☐ demande divisionnaire

☐ certificat d'utilité ☐ transformation d'une demande de brevet européen

☐ demande initiale
☐ brevet d'invention

n° du pouvoir permanent références du correspondant téléphone
MF-BFF990041 01 44 63 4111

Établissement du rapport de recherche

☐ différé ☒ immédiat

Le demandeur, personne physique, requiert le paiement échelonné de la redevance ☐ oui ☐ non

Titre de l'invention (200 caractères maximum)

SYSTEME DE DEVELOPPEMENT RAPIDE D'APPLICATIONS INTERACTIVES.

3 DEMANDEUR (S) n° SIREN code APE-NAF

Nom et prénoms (souligner le nom patronymique) ou dénomination

1. **FRANCE TELECOM**
2. **TELEDIFFUSION DE FRANCE**

Forme juridique

SOCIETE ANONYME
SOCIETE ANONYME

Nationalité (s) 1. **FRANCAISE** 2. **FRANCAISE**

Adresse (s) complète (s)

Pays

1. **6, PLACE D'ALLERAY 75015 PARIS** **FRANCE**
2. **10, RUE D'ORADOUR-SUR-GLANE 75015 PARIS** **FRANCE**

En cas d'insuffisance de place, poursuivre sur papier libre ☐

4 INVENTEUR (S) Les inventeurs sont les demandeurs ☐ oui ☒ non Si la réponse est non, fournir une désignation séparée

5 RÉDUCTION DU TAUX DES REDEVANCES ☐ requise pour la 1ère fois ☐ requise antérieurement au dépôt ; joindre copie de la décision d'admission

6 DÉCLARATION DE PRIORITÉ OU REQUÊTE DU BÉNÉFICE DE LA DATE DE DÉPÔT D'UNE DEMANDE ANTÉRIEURE

pays d'origine numéro date de dépôt nature de la demande

7 DIVISIONS antérieures à la présente demande n° date n° date

8 SIGNATURE DU DEMANDEUR OU DU MANDATAIRE

(nom et qualité du signataire)

CABINET PLASSERAUD
Michel FRECHEDE (CPI N°92-1093)

SIGNATURE DU PRÉPOSÉ À LA RÉCEPTION

SIGNATURE APRÈS ENREGISTREMENT DE LA DEMANDE À L'INPI

DÉSIGNATION DE L'INVENTEUR

(si le demandeur n'est pas l'inventeur ou l'unique inventeur)

N° D'ENREGISTREMENT NATIONAL

990 29 45

DEPARTEMENT DES BREVETS

26bis, rue de Saint-Petersbourg
75800 Paris Cédex 08
Tél. : 01 53 04 53 04 - Télécopie : 01 42 93 59 30

MF/EMA-BFF990041

TITRE DE L'INVENTION :

**SYSTEME DE DEVELOPPEMENT RAPIDE D'APPLICATIONS
INTERACTIVES.**

Les Demanderesses, FRANCE TELECOM et TELEDIFFUSION DE FRANCE,
représentées par
LE(S) SOUSSIGNÉ(S)

**CABINET PLASSERAUD
84, RUE D'AMSTERDAM
F-75440 PARIS CEDEX 09**

DÉSIGNE(NT) EN TANT QU'INVENTEUR(S) (indiquer nom, prénoms, adresse et souligner le nom patronymique) :

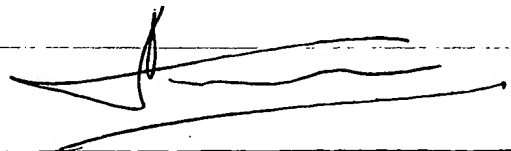
1. HERVET Yves
6, AVENUE GABRIEL FAURE
35235 THORIGNE FOUILLARD
FRANCE
2. CARADO Jean-Antoine
8, RUE JEAN MERMOZ
35520 MELESSE
FRANCE

NOTA : A titre exceptionnel, le nom de l'inventeur peut être suivi de celui de la société à laquelle il appartient (société d'appartenance) lorsque celle-ci est différente de la société déposante ou titulaire.

Date et signature (s) du (des) demandeur (s) ou du mandataire

Paris, le 9 juillet 1999.

**CABINET PLASSERAUD
Michel FRECHEDE (CPI N°92-1093)**



SYSTÈME DE DÉVELOPPEMENT RAPIDE
D'APPLICATIONS INTERACTIVES

La présente invention est relative à un système de
5 développement rapide d'applications interactives.

Le développement très important des systèmes de
télécommunication numérique, par câble, satellite ou ré-
seau hertzien, provoque un accroissement constant et sou-
tenu de la demande d'applications interactives diffusées
10 par ces systèmes de télécommunication numérique, applica-
tions interactives associées ou non à des programmes de
télévision.

Ces applications peuvent concerner des services
aussi divers que prévisions météorologiques, gestion de
15 portefeuilles boursiers, applications de type magazine TV,
radio ou analogue.

D'une manière générale, et afin de permettre l'es-
sor et la pérennité du développement de ces applications,
en raison de la complexité croissante de ces applications
20 et des services fournis par ces dernières, les terminaux
d'accès à ces applications, initialement constitués avant
tout par des récepteurs TV munis d'un module de télécom-
mande, ont évolué vers des terminaux d'ordinateurs munis
de capacités de traitement d'images vidéo et de restitu-
25 tion sonore, appareils plus prosaïquement désignés par
"ordinateurs de salon" dans le domaine grand public.

En raison de cette évolution, les terminaux d'ac-
cès à ces applications étant ainsi dotés de systèmes d'ex-
ploitation de plus en plus puissants, il existe une forte
30 demande pour l'accès à des applications de plus en plus
sophistiquées.

En raison du caractère très divers, voire disparate, des terminaux d'accès à ces applications interactives, ce type d'applications pouvant être accessible par l'intermédiaire de systèmes très différents tel que
5 *OPEN TV* commercialisé par la société *OPEN TV Incorporated* aux Etats-Unis, *Media Highway*, distribué en France par la société *CANAL +*, *Mheg* système ouvert répondant au système de normalisation *ISO*, et système *JAVA* dédié à la télévision, en cours d'établissement, le développement d'applications
10 interactives est très complexe, dans la mesure où un tel développement ne peut être envisagé qu'en fonction du système d'accès mis en œuvre.

En d'autres termes, dans l'optique d'un tel développement, ce dernier ne peut être réalisé que pour un
15 système d'exploitation donné correspondant à celui du système d'accès final utilisé.

Dans de telles conditions, les applications interactives précitées sont confrontées, pour chaque système d'exploitation considéré, aux contraintes ci-après :

20 - définition de la structure de l'application interactive considérée et de son modèle logique, une application interactive étant composée de scènes, chaque scène étant elle-même composée d'objets virtuels formés par des objets ou composants divers en permettant la représentation ;
25

- définition des caractéristiques du terminal d'accès, ou plate-forme cible permettant l'accès à ou l'exécution de cette application interactive, ces caractéristiques pouvant comporter, notamment, la taille mémoire vive RAM disponible, les fontes accessibles et les palettes de couleurs disponibles ;
30

- discrimination entre éléments ou objets permanents, téléchargés avec l'application, et nécessaires au fonctionnement minimal de l'application, des éléments temporels, nécessaires à un instant donné, diffusés cycliquement ;

- mise à jour d'une application interactive ou d'une ou plusieurs scènes à partir de données internes ou externes au système d'accès ;

- réutilisation des objets, objets virtuels d'une scène ou d'une ou plusieurs scènes elles-mêmes ;

- association aux événements externes au système d'accès, introduits par un clavier, un pavé numérique ou un boîtier de télécommande de téléviseur, un modem ou une carte à microprocesseur, et aux événements internes au système d'accès tels que des temporisations, un langage de commande suffisant pour conférer à l'application son caractère interactif ;

- contrôle de la construction d'une scène, de sa représentation et des enchaînements entre scènes successives.

Alors que les applications interactives précitées sont actuellement construites pour le et dédiées au système d'exploitation de la plate-forme ou du système d'accès, ces applications doivent, à l'heure actuelle, être construites dans le cadre strict correspondant.

La présente invention a pour but de remédier aux inconvénients précités par la mise en œuvre d'un système de développement d'applications interactives non dédiées au système d'accès à ces applications.

En particulier, dans ce but, un objet de la présente invention est la mise en œuvre d'un système de déve-

loppement d'applications interactives permettant d'ouvrir le média TV, c'est-à-dire le système de télécommunication par la télévision, à tout type d'application interactive liée ou non aux programmes TV télédiffusés.

5 En particulier, et en raison de l'absence de lien entre les applications interactives, développées grâce au système de développement objet de la présente invention, et le ou les systèmes d'accès à ces applications, un autre objet de cette dernière est la mise en œuvre d'un système
10 de développement d'applications interactives permettant le développement d'applications interactives natives et la traduction de ces applications interactives natives en applications interactives dédiées à l'un ou l'autre des systèmes d'exploitation utilisé par le système ou plate-forme
15 d'accès à ces applications interactives dédiées.

En référence à l'objet précédemment cité, un autre objet de la présente invention est enfin, grâce au processus de traduction d'applications interactives natives développées grâce au système de développement objet de la
20 présente invention, la mise en œuvre d'un processus de traduction de ces applications interactives natives en des applications interactives dédiées à des médias autres que les médias TV, en particulier les médias de transmission de messages ou documents en réseau tels que documents
25 HTML.

Le système de développement d'applications interactives distribuées sur un canal de transmission numérique, objet de la présente invention, ces applications interactives étant installées en un point du réseau de
30 transmission et consistant en une succession de scènes thématiques incorporant des images en mode point, des ima-

ges vidéo, des séquences sonores et des informations sous forme de fichiers texte représentés au niveau d'au moins un moniteur d'affichage, le caractère interactif de ces applications étant produit par l'intermédiaire d'une fonctionnalisation spécifique de l'interface de commande d'un module de commande du moniteur d'affichage muni d'un système d'exploitation, l'ensemble moniteur d'affichage, module de commande et système d'exploitation constituant une plate-forme d'exécution, est remarquable en ce qu'il comporte au moins un éditeur électronique permettant la simulation, d'une part, de l'écran d'affichage de ce moniteur d'affichage, et, d'autre part, de l'interface de commande d'un module de commande associé à ce moniteur d'affichage. Il comporte en outre un module générateur d'objets virtuels, spécifiques à l'application considérée, chaque objet virtuel étant constitué par des objets ou composants choisis parmi au moins des images en mode point, des images vidéo, des séquences sonores et des informations sous forme de fichiers textes et susceptible d'être édité, affiché et représenté par l'éditeur électronique pour former au moins une scène constitutive d'une application interactive considérée.

Un module de commande de mise en corrélation d'une pluralité de couples événement/action est prévu, un événement étant défini comme la rupture d'un équilibre d'état du système d'exploitation et une action étant définie comme une initialisation respectivement une modification d'au moins un des paramètres d'affichage et de représentation d'une ou plusieurs scènes constitutives de cette application. Un module de mémorisation de l'ensemble de ces couples événement/action mis en corrélation et constituant

l'application interactive, au moins sous forme native, est prévu.

Le système de développement d'applications interactives, objet de la présente invention, trouve application à la production d'applications interactives distribuées sur des canaux de transmission tels que les canaux de télévision et les systèmes multimédia tels que le W.W.W., les applications interactives étant, dans ce cas, produites sous forme de documents HTML, de langage interprété JAVA connues sous le nom de "*JAVA script*" d'applications JAVA ou "*Applet JAVA*" ou d'un document au format exécutable par un logiciel de représentation audio ou vidéo, désigné par "*pluggin*" spécifique.

Il sera mieux compris à la lecture de la description et à l'observation des dessins dans lesquels :

- la figure 1a représente aux points 1) à 5) un schéma synoptique illustrant la gestion des documents dans le cas des plates-formes d'accès de type OpenTV, Mheg, JavaTV, applications HTML, application spécifique pour W.W.W par l'intermédiaire d'un logiciel de représentation audio ou vidéo désigné par "*pluggin*" respectivement ;

- la figure 1b représente, à titre illustratif, un schéma synoptique du système de développement d'applications interactives, objet de la présente invention ;

- la figure 1c représente, sous forme d'une structure arborescente, les éléments constitutifs d'une application interactive permettant la construction de tout type d'application interactive grâce au système de développement objet de la présente invention tel que représenté en figure 1b ;

- la figure 1d représente, sous forme d'un organigramme, le format logique d'une application interactive développée grâce au système de développement objet de la présente invention, ce format logique introduisant les relations fonctionnelles entre les différentes structures de données permettant le développement de cette application interactive ;

10 - la figure 2 représente, à titre d'exemple non limitatif, l'édition d'une interface de commande d'un module de commande de type boîtier de télécommande de téléviseur ;

- les figures 3a₁ à 3a₃ représentent, à titre d'exemple non limitatif, un outil logiciel de création de macro-commandes d'instructions ;

15 - les figures 3b₁ et 3b₂ représentent, à titre d'exemple non limitatif, un outil logiciel de création d'objets virtuels ;

20 - la figure 3c représente, à titre d'exemple non limitatif, un outil logiciel d'édition des médias, un média étant défini dans le contexte de développement d'une application interactive grâce au système de développement objet de la présente invention, par un moyen déterminé de représentation d'un objet ou composant constitutif d'au moins une scène de cette application interactive ;

25 - la figure 3d représente, à titre d'exemple non limitatif, un outil logiciel d'édition des fonds de scène ;

30 - la figure 3e représente, à titre d'exemple non limitatif, un outil logiciel d'édition de phrases, sous forme de fichiers texte affectés soit à une scène, soit à l'application interactive contenant cette scène ;

- la figure 3f représente, à titre d'exemple illustratif, un outil logiciel d'édition de variables, ces variables correspondant à un contenu texte ou numérique ;

5 - la figure 3g représente, à titre d'exemple illustratif, un outil logiciel de création de scènes ;

- les figures 3h₁ et 3h₂ représentent, à titre d'exemple non limitatif, un outil logiciel de création d'options ;

10 - la figure 3i représente, à titre d'exemple illustratif, un outil logiciel d'édition des attributs d'une application ;

- les figures 3j₁ et 3j₂ représentent, à titre d'exemple non limitatif, un outil logiciel de consultation de tables communes de données spécifiques au système de plate-forme cible d'accès à l'application interactive développée ;

15 - la figure 4a représente, à titre d'exemple purement illustratif, un organigramme fonctionnel de création d'une application interactive pour un système d'exploitation du type *OPEN TV* à partir des structures de données obtenues grâce à la mise en œuvre du système de développement objet de la présente invention tel que décrit en liaison avec les figures précédentes, ces structures de données constituant une application interactive native,

20 susceptible d'être traduite en l'un ou l'autre des formats du système d'exploitation du système ou plate-forme cible d'accès à cette application interactive ;

25 - la figure 4b représente, à titre purement illustratif, une page écran de gestion du processus de création représenté en figure 4a ;

30

- la figure 5a représente, à titre d'exemple purement illustratif, un module d'exploitation et d'acquisition des objets ou composants constitutifs de chaque scène d'une application interactive plus particulièrement adapté à la diffusion cyclique de cette application interactive sur un canal de télévision, ce module d'exploitation et d'acquisition permettant d'optimiser à la fois la quantité d'espace mémoire nécessaire et le temps d'accès à cette application interactive au niveau d'une plate-forme ou d'un système cible d'accès de type déterminé.

- la figure 5b représente, à titre illustratif, un organigramme fonctionnel représentatif de l'ensemble des séquences constitutives d'une application interactive arbitraire ;

Une description plus détaillée du système de développement d'applications interactives, objet de la présente invention, sera maintenant donnée en liaison avec les figures 1a à 1c et les figures suivantes.

D'une manière générale, on rappelle, d'une part, que les applications interactives sont installées en un point du réseau de transmission numérique afin de permettre une transmission de ces applications interactives, soit sur appel de l'utilisateur dans le cas où ces applications interactives sont transmises par l'intermédiaire du W.W.W., pour *World Wide Web*, sous forme de documents HTML, soit télédiffusées de manière cyclique lorsque le canal de transmission numérique est un canal de télévision par exemple. On indique, de manière plus spécifique, que ces données cycliques peuvent également contenir des documents HTML et des applications ou langages interprétés JAVA, telles que définies précédemment. Dans ce cas, les

applications interactives précitées sont installées en tête du réseau de télédiffusion, alors que dans le cas de la transmission de documents HTML, ces applications sont par exemple implantées au niveau d'un site W.W.W et d'un

-5- - serveur par exemple. - - - - -

La notion de document est, de manière générale, explicitée en fonction des différentes plates-formes d'accès susceptibles d'être utilisées en référence à la figure 1a, aux points 1) à 5), pour les plates-formes de type

10 OpenTV, Mheg, et JavaTV, applications HTML proprement dites et applications pour logiciel de représentation audio ou vidéo désigné par "pluggin" spécifique pour le W.W.W. tel que "Roadster", "Neuron" pour plate-forme d'accès INTERNET. Le document est géré par le système d'exploitation

15 OS OpenTV par l'intermédiaire d'un pilote ADR, par un interpréteur Mheg, par une machine virtuelle JavaTV, par un interpréteur HTML et par un logiciel de type client/serveur, encore désigné par "player", et un navigateur NETSCAPE ou INTERNET EXPLORER respectivement.

20 D'autre part, on rappelle également que chaque application interactive consiste en une succession de scènes thématiques pouvant incorporer des images en mode point, des images vidéo, des séquences sonores et des informations sous forme de fichiers texte par exemple. Lors de

25 l'accès à partir d'un système ou d'une plate-forme d'accès à ces applications interactives, ces scènes sont représentées au niveau d'au moins un moniteur d'affichage incorporé dans le système ou la plate-forme d'accès précité. Par représentation d'une ou plusieurs scènes et, en définitive, de l'application interactive considérée, on entend

30 la restitution à l'utilisateur, non seulement de l'ensem-

ble des informations vidéo ou images par affichage au niveau de ce moniteur, mais également la restitution sonore de tout fichier correspondant par l'intermédiaire de circuits audio équipant le système ou la plate-forme d'accès.

5 - - - - - Le caractère interactif des applications précitées - - - - -
est produit par l'intermédiaire d'une fonctionnalisation spécifique de l'interface de commande d'un module de commande du moniteur d'affichage muni, bien entendu, d'un système d'exploitation.

10 Ainsi, l'ensemble moniteur d'affichage, module de commande et système d'exploitation constitue un système ou plate-forme d'exécution ou d'accès à l'application interactive considérée.

15 On rappelle en particulier que le système d'exploitation considéré peut être très divers selon la plate-forme d'accès utilisée. Ces systèmes d'exploitation peuvent notamment être constitués par les systèmes MacOS, PC pour *Personal Computer*, UNIX, LINUX ou autres. En particulier, lorsque la plate-forme d'accès utilisée correspond
20 au système *OPEN TV* précédemment mentionné dans la description, on indique que le système d'exploitation considéré est un système d'exploitation réduit ou dédié à l'application interactive considérée. Dans un tel cas, le module de commande est le plus souvent constitué par un module de
25 commande ou de télécommande universelle à boutons, permettant de commander le téléviseur muni du moniteur d'affichage de type télévision. On comprend en particulier que le système d'exploitation dédié, dans ce cas, peut correspondre à l'affectation de certaines fonctions aux boutons
30 du boîtier de télécommande en fonction des nécessités de l'application.

Dans le cas où l'application interactive concerne les documents HTML transmis sur le W.W.W. et lorsque la plate-forme d'accès est constituée par un micro-ordinateur, le système d'exploitation considéré n'est autre que le système d'exploitation de ce micro-ordinateur, bien entendu adapté à la gestion des documents HTML. Dans ce dernier cas, le système d'exploitation global est beaucoup plus puissant que dans le cas précédent.

Ainsi que représenté en figure 1b, le système de développement d'applications interactives, objet de la présente invention, est construit sur la base d'un micro-ordinateur ou d'une station de travail comportant tous les éléments essentiels de ces derniers, à savoir une unité centrale de calcul CPU, un moniteur d'affichage M, une mémoire de travail RAM et une mémoire morte ROM ainsi que des organes d'entrée/sortie périphériques tels que par exemple un clavier Kb, un lecteur de disque souple Fl, une mémoire de masse de type disque dur HD et tout moyen d'intercommunication lorsque ce micro-ordinateur est connecté en réseau. Ces éléments ne seront pas décrits en détail car ils correspondent à des éléments classiques équipant tout système de micro-ordinateurs ou de station de travail.

En référence à la figure 1b, on indique que le système de développement d'applications interactives, objet de la présente invention, est remarquable en ce qu'il comprend un éditeur électronique, noté E, permettant la simulation de l'écran d'affichage du moniteur d'affichage, ainsi que de l'interface de commande d'un module de commande associé à ce moniteur d'affichage. Sur la figure 1b, on a représenté, à titre d'exemple non limitatif, la simu-

lation d'un boîtier de commande de télévision par l'intermédiaire du moniteur M et bien entendu de l'éditeur E. De même, sur la figure 1b, on a représenté la simulation de l'écran d'affichage du moniteur d'affichage en un moniteur d'affichage simulé MS par l'intermédiaire de l'éditeur E, ce moniteur d'affichage simulé MS étant réputé représenter une scène d'une application interactive donnée. Ainsi, le module de commande simulée, noté MCS, et l'écran d'affichage simulé du moniteur d'affichage simulé MS sont représentés sur le moniteur M du micro-ordinateur abritant le système de développement d'applications interactives, objet de l'invention.

En outre, ainsi que représenté sur la même figure 1b, le système précité comporte un module générateur d'objets virtuels, noté GOV, spécifique à l'application considérée. Chaque objet virtuel est constitué par des composants choisis parmi au moins des images en mode point, des images vidéo, des images 3D, des séquences sonores et des informations sous forme de fichiers texte qui sont susceptibles d'être édités, affichés et représentés par l'éditeur électronique E pour former au moins une scène constitutive de l'application.

Par objet virtuel, on entend des collections d'objets ou composants constituées par des médias selon la définition précédemment donnée dans la description. Ainsi, lorsque l'objet ou composant est constitué par une séquence sonore, c'est-à-dire par un fichier numérique permettant la reproduction d'une séquence sonore, ce dernier peut être représenté par les circuits audio de la plateforme ou système d'accès à l'application interactive considérée. Lorsque l'objet ou le composant est constitué par

une image en mode point ou une image vidéo, cet objet peut être également représenté par affichage, par une image en mode point ou une image vidéo, c'est-à-dire une image munie de ses caractéristiques de balayage spécifique, au niveau du moniteur et du moniteur simulé MS, alors que lorsque l'objet ou composant est constitué par un fichier texte, celui-ci peut être affiché en mode texte par exemple sur le moniteur M et bien entendu sur le moniteur d'affichage simulé MS.

10 En outre, ainsi que représenté sur la figure 1b, le système de développement, objet de l'invention, comprend un module de commande, noté COR, de mise en corrélation d'une pluralité de couples événement/action relatifs aux scènes constitutives de l'application interactive précitée.

15 De manière plus spécifique, on indique qu'un événement est défini comme la rupture d'un équilibre d'état du système d'exploitation par une entrée externe dans le cadre de l'exécution de l'application interactive, alors qu'une action est définie comme une initialisation ou, le cas échéant, une modification d'au moins l'un des paramètres d'affichage et de représentation d'une ou plusieurs scènes constitutives de l'application interactive considérée. Bien entendu, une action est réputée être consécutive à un événement ou réciproquement en raison du caractère interactif de l'application considérée.

25 Enfin, un module de mémorisation, tel qu'un disque dur HD, de l'ensemble des couples événement/action mis en corrélation est prévu, cet ensemble constituant en fait l'application interactive, au moins sous forme native, ainsi qu'il a été mentionné précédemment dans la descrip-

tion. Par application interactive native, on désigne l'ensemble des couples événement/action mis en corrélation grâce au système de développement, objet de la présente invention, cet ensemble des couples événement/action étant
5 ainsi réalisé par mise en correspondance bi-univoque logique
entre événement et action indépendamment du format d'accès imposé par le système ou plate-forme d'accès à l'application interactive considérée du fait des caractéristiques physiques et formelles de ce système ou de cette
10 plate-forme d'accès.

Ainsi, sur la figure 1b, on a représenté, d'une part, les composants physiques de calcul et de mémorisation du micro-ordinateur tels que unité de calcul CPU, mémoire morte ROM, mémoire vive de travail RAM et une
15 mémoire de masse HD, l'ensemble étant désigné par la référence C et intégré au micro-ordinateur abritant le système de développement, objet de la présente invention, ainsi que les éléments actifs de ce dernier, c'est-à-dire éditeur électronique E permettant la simulation de l'écran
20 d'affichage du moniteur d'affichage simulé MS et de l'interface de commande, ou module de commande, MCS, associée à ce moniteur d'affichage accompagnés du module générateur d'objets virtuels GOV et du module de commande de mise en corrélation COR. Sur la figure 1b, le module générateur
25 d'objets virtuels GOV et le module de commande de mise en corrélation des couples événement/action COR sont réputés réalisés sous forme d'un logiciel pilote chargé en mémoire de travail RAM, soit à partir du disque dur HD, soit le cas échéant à partir de la mémoire morte ROM. L'ensemble
30 est contenu dans le boîtier B.

Préalablement à la description du module générateur d'objets virtuels et du module de commande de mise en corrélation de couples événement/action, une description de la structure de données permettant la définition d'une application interactive grâce au système de développement, objet de la présente invention, sera donnée en liaison avec la figure 1c.

En référence à la figure précitée, on indique qu'une application interactive est définie par un ensemble de listes informatiques définissant, pour cette application interactive, les éléments structurels communs.

Ainsi que représenté sur la figure 1c, cet ensemble de listes peut comporter :

- une liste des données, cette liste des données pouvant consister en des fichiers externes numériques ou des fichiers texte relatifs à l'application interactive considérée.

La liste des données permet de définir le contexte de l'application ou d'un groupe d'applications données telles que par exemple une application interactive de service météo, de service de résultats de jeu tel que le loto, ou d'un horoscope ou de consultation d'un thème astral. Les données sont alors des données physiques, telles que par exemple celles relatives à une région pour un service météo ou analogue ;

- une liste d'attributs de l'application, notée AP ;
- une liste des objets communs OC pouvant comporter une description de ces objets ;
- une liste des événements/actions relative à l'application, c'est-à-dire des événements/actions mis en corré-

lation, liste notée AA, pouvant comporter une liste descriptive de ces événements/actions ;

- une liste des fonds de scène ou liste des *backgrounds*, notée BG ;

- 5 - - - - une liste des scènes, la liste des scènes comportant des références de scènes ordonnées permettant de représenter la séquence des scènes pour constituer une application interactive considérée ;
- une liste des médias ME utilisés, la notion de médias correspondant à celle précédemment donnée dans la description ;
- 10 - une liste des variables VA ;
- une liste des phrases PH, les phrases étant considérées comme des fichiers texte ;
- 15 - une liste des palettes PA de couleurs pour l'application considérée, et, enfin,
- une liste des couleurs CO effectivement utilisées.

En ce qui concerne la liste des scènes, et selon un mode de réalisation particulièrement avantageux, non limitatif, du système de développement, objet de la présente invention, on indique que chaque scène peut être définie par une liste de scènes comportant par exemple :

- au moins un élément relatif à un fond de référence ;
- un élément relatif à un attribut de scène SC ;
- 25 - une liste des objets ou composants constitutifs de la scène ;
- une liste des événements/actions de la scène, notée AS ;
- un attribut de curseur, noté OP, cet attribut de curseur définissant notamment la position d'origine du
- 30

curseur dans la scène afin de permettre l'utilisation interactive de la scène par l'utilisateur ;

- une liste des phrases PS.

5 Une description plus détaillée du format logique d'une application interactive native développée grâce au système de développement objet de la présente invention, sera maintenant donnée en liaison avec la figure 1d.

10 D'une manière générale, on indique que ce format logique introduit les relations fonctionnelles entre les différentes structures de données permettant le développement d'une application interactive considérée, au moins sous forme native.

15 Sur la figure 1d précitée, on indique que tous les blocs rectangulaires représentent de manière non limitative des structures de données remarquables mises en œuvre par chaque application numérique développée conformément au système de développement rapide d'applications interactives, objet de la présente invention, l'ensemble de ces structures de données et les relations fonctionnelles pré-

20 citées, et en particulier les couples "EVENEMENT" / "ACTION" précédemment mentionnés dans la description permettant la définition de l'application interactive native considérée.

25 Ainsi, sur la figure 1d, on a représenté les différentes structures de données représentées par des rectangles, ces rectangles étant reliés par des flèches indiquant la relation logique entre les structures de données correspondantes. Les structures de données peuvent être représentées sous toute forme de langage, notamment

30 les langages objets C++ et JAVA.

Les différents rectangles portant les références 100 à 119 désignent les structures de données ci-après :

- 100 : palette de couleurs
- 101 : couleur choisie
- 102 : couleur choisie dans la palette considérée
- 103 : type de média utilisé
- 104 : forme du média
- 105 : type de fond ou *background*
- 106 : fond ou *background*
- 107 : type de variable VA-type
- 108 : variables, les variables étant soit de type texte, soit de type numérique
- 109 : "EVENEMENT"
- 110 : "ACTION"
- 111 : application
- 112 : type d'objet, désigné par OBJ_TYPE,
- 113 : objet d'interface
- 114 : couple "EVENEMENT/ACTION" mis en corrélation
- 115 : phrases
- 116 : scène
- 117 : options
- 118 : type d'affichage, désigné par Disp_Type
- 119 : condition.

Les relations logiques entre structures de données précitées telles que représentées sur la figure 1d sont données au tableau 1 introduit en annexes.

Dans la présente description, les tableaux 1 à 28 relatifs à des structures de données sont introduits en annexes.

Dans le tableau 1, les flèches représentent le sens des flux de données entre structures de données.

Des indications plus générales, relatives à l'organisation de la structure des données précitée, seront
5 maintenant introduites ci-après dans la description.

D'une manière générale, pour chaque plate-forme d'accès, un pilote interpréteur des données de l'application dans le langage du moteur interactif est prévu. Les données logiques et les données dépendant de chaque plate-
10 forme sont structurées de manière distincte.

Les données sont agencées sous forme de structures de tables communes relativement aux plates-formes PL, aux fontes FO, aux actions AC, aux événements EV, et aux données DO décrivant et caractérisant d'une façon originale
15 l'environnement de la plate-forme ou du terminal d'accès à l'application interactive considérée.

Ainsi, c'est dans le cadre des structures de données précitées et de leur relation logique, telle que représentée en figure 1d, que l'ensemble des éléments du
20 système de développement d'applications interactives, objet de la présente invention, permet d'effectuer un tel développement.

A titre d'exemple non limitatif, les tables de données relatives aux plates-formes PL peuvent présenter
25 la structure ci-après telle que représentée au tableau 2.

Dans le tableau précité, on indique que les données relatives aux tables plates-formes PL peuvent comprendre avantageusement les variables ci-après : nombre maximum de caractères pour les textes, fixé par exemple à
30 255, à titre d'exemple non limitatif, et pour chaque plate-forme PL : identifiant de la plate-forme, nom de la

plate-forme, nombre de couleurs maximum de la palette, ce nombre maximum, dans le cadre du système *OPEN TV* étant limité à 15, mémoire maximale en kilo-octets pour le chargement des médias, espace disponible pour le média sur le disque dur HD par exemple, nombre maximum de médias par disque ou par module de mémorisation, caractère de fin de ligne pour les textes, mode de sécurité de la plate-forme et, le cas échéant, un paramètre de 0 à 255 indiquant l'efficacité de la transparence, c'est-à-dire l'opacité de la couleur de transparence.

En ce qui concerne la structure de données relative aux actions et aux tables d'actions AC, une action se caractérisant par plusieurs paramètres et ne pouvant être déclenchée que par un événement, la structure de données correspondante est donnée dans le tableau 3.

La structure de données précitée représentée au tableau précédent peut comporter, pour chaque action, une variable d'identification de l'action, de type de l'action, c'est-à-dire ce sur quoi celle-ci porte, de nom de l'action, et pour chaque plate-forme PL, au moins une variable booléenne dont la valeur est vraie si cette action est disponible pour cette plate-forme, une variable de description de l'action si celle-ci est disponible pour la plate-forme PL considérée, par exemple le nom de la fonction ou de la procédure à appeler.

En ce qui concerne les fontes de caractères disponibles FO sur une plate-forme ou terminal d'accès considéré, la structure de données est représentée au tableau 4.

Pour les fontes disponibles, on indique que la structure de données FO, pour chaque fonte, peut contenir un identifiant de la fonte, le nom de la fonte et, pour

chaque plate-forme, une référence de la fonte pour la plate-forme considérée, ou plate-forme cible, telle que, en fonction du système d'exploitation de celle-ci, *OPEN TV*, Internet : nom de fonte + taille et de type MAC OS comprenant un identificateur de fonte ainsi que le nom de fonte + la taille.

En ce qui concerne les structure de données relatives aux événements, pour chaque plate-forme, on rappelle que les événements peuvent être internes ou externes lorsqu'ils sont provoqués par la sélection d'une touche sur un clavier par un utilisateur par l'introduction d'une carte à mémoire ou la réception d'un message par exemple, ainsi que mentionné précédemment dans la description.

Dans ces conditions, la structure des données relative à la liste des événements EV pour chaque plate-forme est donnée par le tableau 5.

Relativement aux événements, on indique que pour chaque événement, la structure de données peut être définie par un identifiant de l'événement, un type de l'événement selon ce qui engendre l'événement, un nom de l'événement, ainsi que par des paramètres relatifs au type d'événement considéré.

En outre, pour chaque plate-forme PL, peuvent être définies une variable booléenne à la valeur vraie si l'événement est disponible pour cette plate-forme et, si cet événement est disponible, une variable de description de l'événement pour la plate-forme PL considérée.

En ce qui concerne les données DO décrivant et caractérisant l'environnement de la plate-forme terminale, leur structure de données correspondante peut correspondre à celle donnée dans le tableau 6.

Pour chaque donnée, cette structure de données relative aux données DO peut contenir un identifiant de la donnée, désigné par DATA-ID, une origine de la donnée, un type de la donnée et un nom de la donnée. De même que dans le cas de la structure de données relative aux événements, pour chaque plate-forme PL, peuvent être définies une variable booléenne à la valeur vraie si la donnée est disponible pour cette plate-forme, ainsi qu'une description de cette donnée pour la plate-forme considérée si cette donnée est disponible. Cette description peut consister en le nom de la fonction API pour *Application Protocol Interface* qui retourne la donnée précitée.

Le tableau 6 précédemment mentionné peut contenir des données liées à différentes API, par exemple des données liées à n systèmes de signalisation d'une voie de données, des données contenues dans une carte à mémoire ou encore des données relatives aux émissions de télévision à travers le système d'information défini par la norme DVB, pour *Digital Video Broadcasting*.

D'autres structure de données supplémentaires peuvent être définies pour chaque application ou groupe d'applications, de manière à permettre la mise à jour des données propres à une application par l'intermédiaire d'un automate par exemple. Cette structure de données supplémentaire ne sera pas décrite en détail.

D'une manière générale, et selon un aspect particulièrement remarquable du système de développement d'applications interactives, objet de la présente invention, l'ensemble des couples événement/action constitue l'application interactive, au moins sous forme native précédemment mentionnée dans la description, dans la mesure où cet

ensemble définit la suite des enchaînements nécessaires à la représentation de cette application interactive.

Dans ce but, le module COR de commande de mise en corrélation d'une pluralité de couples événement/action

5 comporte un module générateur d'une série d'instructions présentant une structure de contrôle de type liste de requêtes pour un événement donné vis-à-vis duquel une relation de type IF (condition) sur toute variable booléenne B_i appartenant à un ensemble de variables booléennes

10 $[B_0, B_1, \dots, B_j, B_N]$ est vraie *THEN* (alors), l'action est alors réalisée, *EVENEMENT* désignant une variable représentative d'un événement donné et *ACTION* désignant une variable représentative d'une action conditionnée à l'ensemble des variables booléennes précité.

15 On rappelle que la variable "*EVENEMENT*" constitue pour chaque couple événement/action une rupture d'équilibre d'état du système d'exploitation, alors que l'action constitue une initialisation ou une modification d'au moins un des paramètres d'affichage et de représentation

20 d'une ou plusieurs scènes constitutives de l'application interactive considérée.

Ainsi, l'ensemble des couples événement/action est organisé selon une suite de lignes de macro-instructions comprenant chacune un champ de position d'exécution, un

25 champ d'événement, un champ d'action, un champ de liste de conditions, et un champ variables booléennes d'arrêt.

Le jeu de macro-instructions précité constitue ainsi un langage de très haut niveau de conception, adapté à l'audiovisuel, quelque soit le moteur interactif utilisé

30 et permettant une mise en œuvre particulièrement originale.

D'une manière générale, l'ensemble des événements peut être constitué en une table des événements. Une table des événements est donnée dans le tableau 7.

Dans le tableau 7 précité, chaque événement comporte une désignation mnémotechnique, un numéro d'identification de 1 à 18, le nombre 18 n'étant bien entendu pas limitatif, une référence de type, une liste de paramètres et des commentaires d'événements, en particulier de l'origine de l'événement considéré.

10 Les événements ci-après sont ainsi introduits :

- 1 : Keynav,
- 2 : KeyNum,
- 3 : KeyOK,
- 4 : KeyHome,
- 15 - 5 : KeyBack,
- 6 : AppliStart,
- 7 : SceneOpen,
- 8 : TimerExpired,
- 9 : SceneClosed,
- 20 - 10 : event1,
- 11 : event2,
- 12 : event3,
- 13 : event4,
- 14 : event5
- 25 - 15 : MpegAff,
- 16 : InsertCard,
- 17 : RemoveCard,
- 18 : Receptmodem.

Les commentaires correspondent à un commentaire de la définition mnémotechnique de l'événement donné précédemment

30

en référence au numéro d'identification précité, ainsi que mentionné dans le tableau 7 précédent.

De la même manière, l'ensemble des actions peut être organisé sous forme d'une table des actions donnée dans le tableau 8.

De même que pour les événements, chaque action comprend une définition mnémotechnique de l'action, un numéro d'identification, un type de l'action, un nombre de paramètres, et des commentaires relatifs à l'action considérée. Dans le tableau 8, 21 actions spécifiques ont été introduites de manière non limitative, accompagnées de deux actions auxiliaires correspondant à des opérations arithmétiques dans les conditions qui seront données ci-après :

- 15 - 1 : Setbg,
- 2 : PlayS,
- 3 : PlayA,
- 4 : Putlnto,
- 5 : AddTo,
- 20 - 6 : GoScene,
- 7 : GoBack,
- 8 : DesigneOpt,
- 9 : LounchApp,
- 10 : QuitApp,
- 25 - 11 : ShowObj,
- 12 : HideObj,
- 13 : SetTimer,
- 14 : Selectserv,
- 15 : StopTimer,
- 30 - 16 : Mute,

- 17 : Unmute,
- 18 : gl_actn,
- 19 : ReadUA,
- 20 : Phone,

5 - - - - - 21 : Nothing, - - - - -

Les opérations arithmétiques sont désignées par Addto, Multo et Divto respectivement et correspondent à la multiplication, respectivement à la division entre deux entiers numériques. D'autres opérateurs peuvent être prévus, tels que Modulo, reste de la division entière, soustracto soustraction, pour des opérations de calcul, ou Speak Writer pour la gestion d'un module de synthèse de la parole, afin d'introduire la notion de restitution d'une voix à partir d'un fichier texte et d'une voix.

15 Les commentaires des actions 1 à 21 correspondent à la désignation mnémotechnique telle que donnée dans le tableau 8 précédent.

Ainsi, en référence à la structure de chaque ligne de macro-instructions comprenant chacune un champ de position d'exécution, un champ événement, un champ options, un champ de liste de conditions et un champ variable booléenne d'arrêt, on indique que, en référence à la position d'exécution, les lignes de macro-instructions sont exécutées séquentiellement en partant de la position 1. Ainsi, en référence à la table des événements, Tableau 7, et à la table des actions, Tableau 8, l'action est exécutée sous contrôle de la structure de contrôle précédemment mentionnée dans la description.

30 Cette structure de contrôle peut être résumée de la façon ci-après. Si (IF) la condition est réalisée,

alors (THEN) l'action de la ligne de macro-instructions est alors exécutée.

En ce qui concerne la condition, on indique que cette condition peut consister en une comparaison d'égalité stricte, de supériorité, d'infériorité ou égalité, supériorité ou égalité entre deux valeurs numériques ou entre deux valeurs contenant des chaînes de caractères.

Dans le cas où la variable booléenne d'arrêt n'est pas à la valeur vraie, le pilote exécute la ligne suivante. Sinon, le pilote arrête l'exécution de l'ensemble des lignes de macro-instructions constitutives de l'ensemble des couple événement/action mis en corrélation.

On indique que le nombre de lignes d'exécution, c'est-à-dire de nombre de lignes de macro-instructions, n'est pas limité.

Une description plus détaillée de la structure des données d'une application interactive quelconque sera maintenant donnée dans la description.

En référence à la figure 1c, on rappelle qu'une application interactive est composée d'une ou plusieurs scènes.

Selon un aspect remarquable du système de développement d'applications interactives, objet de la présente invention, les structures de données relatives à une application interactive déterminée sont définies sur deux niveaux distincts :

- un niveau application proprement dit, définissant la structure de données commune à toutes les scènes ;
- un niveau scène définissant la structure de données propre à une scène déterminée.

Chaque application comporte une structure caractérisant l'identification d'une application, structure de données désignée par AP.

La structure de données AP précitée est donnée
 5 dans le tableau 9.

Pour chaque application AP donnée, les attributs de l'application sont définis par des variables relatives à la description de l'application, au nom de l'auteur et de mention de droit d'auteur, désignée par mention de Copyright, au nom de l'application, à la version de l'appli-
 10 cation et à la définition de la palette de couleurs utilisée telle qu'une palette de couleurs par défaut ou définie pour une scène déterminée, une variable de données contenant un numéro d'identification de la scène de départ, en l'absence de numéro d'identification de scène de
 15 départ la valeur de défaut et la valeur du menu sont définies par HOME, une variable de couleur de transparence de l'application et, pour chaque plate-forme PL d'accès considérée, une variable de type booléen, laquelle, à la va-
 20 leur vraie, indique si l'application devra être engendrée pour la plate-forme PL considérée.

Afin d'assurer un haut degré de convivialité des applications interactives développées, l'ensemble des pa-
 25 lettes de couleurs utilisées dans une application est géré par une structure de données particulière PA, chaque scène étant susceptible d'utiliser une palette de couleurs.

La structure de données PA définissant une palette de couleurs est composée de n couleurs qui sont choisies parmi une liste des couleurs notée CO. En général, une ap-
 30 plication n'utilise qu'une palette par application inter-

active considérée. La liste PA des palettes utilisées par l'application est définie au tableau 10.

Pour chaque palette de couleurs gérée par l'application interactive considérée, celle-ci comprend une variable identifiant de la palette de couleurs considérée, une variable de nom de la palette pour faciliter l'édition et une liste de 15 couleurs en l'absence de couleur transparente vidéo dans le cas où le système OPEN TV est utilisé, enfin, un identifiant de la couleur associé à une position dans la palette considérée. Les noms des variables correspondants sont indiqués au tableau 10.

En outre, une structure de données spécifique, notée BG, définit le fond d'une scène de l'application interactive considérée. A titre d'exemple non limitatif, on indique que les types de fond peuvent être constitués par un fond de couleur, un fond vidéo, une image fixe telle qu'une image fixe de type MPEG2. D'autres types de fond peuvent être définis tels que un fond de type composé pouvant contenir une image fixe en millions de couleurs, ou un fond vidéo ou un fond de couleur suivant la palette utilisée, ainsi qu'un ensemble d'objets ou composants à contenu visuel tels qu'une image, un graphique de couleur, un texte ou une palette de couleurs.

La structure de données définissant le fond d'une scène BG est donnée au tableau 11.

La liste des fonds utilisée par l'application est définie pour chaque fond engendré par une variable identifiant du fond, une variable de type de fond, une variable de valeur de fond selon le type et une variable spécifique, obligatoire, si le type de fond correspond à celui d'une image de type MPEG par exemple, ainsi enfin qu'une

variable de nom du fond pour faciliter l'édition. Les variables précitées sont définies par une définition mnémotechnique correspondante donnée au tableau 11.

En ce qui concerne la variable de type de fond, celle-ci peut comporter plusieurs sous-variables données au tableau 12, telles qu'une couleur unie, une transparence au signal vidéo, une image mpeg en arrière-plan ou une image en mode point par exemple.

Le contenu des fonds utilisés par une application interactive développée grâce au système de développement rapide, objet de la présente invention, ainsi que le contenu des objets constitutifs de ces scènes, fait appel à des médias pour constituer en fait des objets virtuels ou ensemble de composants précédemment mentionnés dans la description qui, bien entendu, doivent être présents ou seront présents lorsque la scène est composée ou représentée.

En ce qui concerne les médias, on rappelle que ceux-ci, dans le cadre de la mise en œuvre du système de développement d'applications interactives, objet de la présente invention, sont définis comme des ressources externes et peuvent être répertoriées en fonction du lecteur audio ou vidéo transmettant l'information destinée à l'utilisateur suivant les types ci-après :

- Audio : nom de ressource/attribut de codage/taille mémoire ;
- Vidéo : nom de ressource/attribut de codage/taille mémoire ;
- Texte speech : nom de ressource / attribut de vitesse / taille mémoire ;

- Image fixe naturelle : nom de ressource/taille mémoire ;
- Image graphique : nom de ressource/taille mémoire/palette de couleurs.

-----5-----La structure de données relative aux médias ME-----
 n'est autre qu'une liste des médias mise en œuvre pour
 l'application interactive considérée. La structure de données
 correspondante est donnée dans le tableau 13 pour
 chaque média géré par cette application interactive précitée.
 10

Les données précitées comportent une variable
 d'identification du média, une variable de type du média,
 une variable de mise à jour, une variable de type booléenne,
 laquelle, lorsqu'elle est à la valeur vraie, permet
 15 de revenir à la mise à jour, en l'absence de média,
 et, enfin, une variable d'utilisation lorsque le média est
 externe, c'est-à-dire éditable en mode exploitation. En
 outre, pour chaque plate-forme ou terminal d'accès à l'ap-
 plication interactive considérée, plate-forme PL, la
 20 structure de données précitée comporte une variable de ré-
 férence du média pour une plate-forme donnée, une variable
 de structure interne, une variable d'identification de la
 référence, une variable de rappel de la plate-forme de
 cette référence et une variable de format du média réfé-
 25 rencé, une variable de taille physique du média considéré,
 une variable de taille mémoire en vue du préchargement et
 une variable de référence du média pour la plate-forme
 considérée. La plate-forme peut, ainsi que mentionné pré-
 cédemment dans la description, consister en une plate-
 30 forme, c'est-à-dire en un terminal d'accès de type MacOS,
 le chemin d'accès complet sur le disque étant indiqué, ou

de type Internet URL ou de type *OPEN TV*, nom du fichier contenant le média considéré, et enfin une variable de nom de média.

5 Une application interactive développée conformément à la mise en œuvre du système, objet de la présente invention, peut faire souvent appel à des variables dont le contenu peut varier pendant le déroulement de l'application. Il peut s'agir par exemple, pour chaque scène indexée, d'une valeur numérique indexée de la valeur 1 dans
10 une variable de comptage. Ces variables constituant des variables numériques de gestion de l'application, et en particulier du caractère interactif de celle-ci, sont désignées par des variables dont la structure de données VA est donnée dans le tableau 14.

15 En référence au tableau précité, pour chaque variable, il peut être défini au moins une variable de type texte, une variable d'identification de la variable de gestion considérée, une valeur d'origine de la donnée, une valeur de type de variable et une valeur de nom de variable
20 ainsi que des lignes de texte ou nombres entiers correspondants. En ce qui concerne le type de la variable considérée, on indique que celle-ci, en référence au tableau 12, peut contenir toute chaîne de caractères de type "txt" ou un nombre entier de type "num".

25 Les scènes d'une application interactive font appel à la définition d'objets virtuels décrits au niveau de cette application. L'objet virtuel peut être défini comme un conteneur dont les caractéristiques et les attributs seront décrits ultérieurement dans la description en liaison avec la description des structures de données relatives à une scène.
30

Selon un mode d'organisation remarquable des structures de données du système de développement d'applications interactives, objet de la présente invention, ces structures de données peuvent avantageusement

5 une liste des objets, virtuels, d'interface communs à toutes les scènes. Cette liste est désignée par OC et la structure des données correspondante est donnée dans le tableau 15.

En référence au tableau précité, pour chaque objet

10 virtuel commun, la structure des données correspondante peut comporter une variable de description de l'objet commun et une variable de structure unitaire de description de cet objet.

En ce qui concerne la gestion des macro-commandes

15 constitutives du langage de haut niveau descriptif de l'application, on indique que l'ensemble des couples événement/action peuvent être décrits par une structure de données relative aux lignes événement/action attachées à l'application interactive considérée, cette structure de

20 données, pour chaque ligne, et donc pour chaque action de l'application considérée, désignée par structure de données AA, est donnée dans le tableau 16.

Dans la structure de données précitée, les lignes événement/action attachées à l'application interactive

25 peuvent comporter avantageusement une variable de numéro d'ordre de l'action, une variable de description de l'action relative à l'application considérée ainsi qu'une variable de structure unitaire de description d'une action, ainsi qu'il sera décrit ultérieurement dans la descrip-

30 tion.

Une description plus détaillée des structures de données mises en œuvre pour la description d'une scène constitutive d'une application interactive développée grâce à un système de développement rapide d'applications interactives, objet de la présente invention, sera maintenant donnée en liaison avec les tableaux 17 à 25.

Par définition, une application interactive est constituée par une pluralité de scènes, la scène considérée étant désignée par scène courante.

10 Selon un premier aspect remarquable du système de développement d'applications interactives, objet de la présente invention, celui-ci comporte une structure de données d'identification de la scène courante, cette structure de données, notée SC, étant donnée par le tableau 17.

La structure de données précitée contient les attributs propres à la scène courante considérée et peut comporter avantageusement une variable d'identification de la scène. Il existe toujours au moins une scène, une scène de référence, encore désignée par scène d'accueil, correspondant à une page HOME de l'application interactive par exemple, une variable de nom de scène, une variable de palette de couleurs, cette variable pouvant correspondre à la variable palette de couleurs de l'application ou à une variable de palette de couleurs différente, spécifique à la scène, une variable d'identifiant du fond, une variable de position de départ du curseur et, enfin, une variable booléenne, laquelle, lorsque celle-ci est à la valeur vraie, permet le retour à la dernière option de la scène considérée.

D'une manière générale, on indique que la composition bidimensionnelle de chaque scène courante de l'application interactive est effectuée à partir de conteneurs définissant les objets virtuels, ainsi que mentionné précédemment dans la description.

La description d'un objet virtuel, c'est-à-dire du conteneur précité et de ses attributs, sera maintenant donnée en liaison avec les tableaux 18 à 24.

D'une manière générale, on indique que la gestion des objets virtuels met en œuvre une structure de données comportant au moins :

- une table des objets dont la structure de données, désignée par OS, est donnée dans le tableau 18.

Le tableau précité définit une liste des objets d'interface de la scène courante et peut comprendre, pour chaque objet, une variable de description de l'objet et une variable de structure unitaire de description de cet objet.

La description d'un objet comporte en outre une structure de données relative à l'ensemble des couples événement/action pour la scène courante considérée, cette structure de données étant notée AS et décrite par le tableau 19.

Pour chaque action de la scène courante précitée, cette structure de données peut comprendre une variable de numéro d'ordre de l'action relative à la scène, une variable de description de l'action relative à la scène et une variable de structure unitaire de description d'une action.

En outre, et de même que pour le niveau application, la structure de données relative à la définition des

objets virtuels d'une scène peut contenir avantageusement une structure de données définissant une liste de phrases constituant en fait un dictionnaire de la scène. Cette structure de données, notée PS, est décrite dans le tableau 20.

La structure de données précitée constitue une liste des phrases attachées à la scène courante et peut comporter, pour chacune des phrases correspondantes, une variable identifiant de la phrase considérée, une variable origine de la donnée, tel que le nom du pilote générateur de cette donnée et du mode d'acquisition en lecture par exemple, une variable de type de phrase, ce type de phrase étant par exemple de type texte "txt", une variable du nom de la phrase, une variable réservée pour saut de ligne et une variable de ligne de texte.

En ce qui concerne la description de chaque objet virtuel relatif à la scène ou le cas échéant à une application considérée, objet virtuel considéré comme une structure unitaire, une structure de données détaillée permettant la description de ce conteneur ou objet virtuel est prévue afin de décrire l'objet virtuel correspondant, selon le tableau 21.

Chaque objet virtuel, défini selon la structure de données précédente, constitue une structure unitaire permettant de définir la description d'un objet relatif à la scène courante ou, le cas échéant, à une application, et peut comporter par exemple une variable d'identification de l'objet, une variable de niveau d'affichage de cet objet, soit en fond de scène, soit en avant-scène, une variable de type d'affichage d'objet, le type d'affichage étant dit statique lorsque l'objet est un objet relatif à

l'application et pouvant être soit statique, soit dynamique lorsque l'objet est relatif à la scène courante d'une application considérée, une variable d'attribut selon le type d'affichage, une variable de type d'objet, une variable de nom de l'objet, de préférence en l'absence de virgule, et une variable d'attribut de l'objet selon le type. Le conteneur ou objet virtuel précité comporte également une structure de données définissant l'attribut d'affichage de cet objet virtuel défini selon le tableau 22.

10 Cette structure de données définit les valeurs possibles pour l'attribut d'affichage de l'objet virtuel considéré et peut comporter une variable d'affichage direct sans condition, une variable d'affichage par l'intermédiaire des actions de type, c'est-à-dire affichage
15 dynamique ou statique, et une variable d'affichage conditionné au numéro d'option du curseur sur l'écran d'affichage.

 De même, une structure de données relative à la description de l'attribut du contenu de l'objet virtuel ou
20 conteneur est décrite selon le tableau 23.

 La structure de données précitée est constituée par une liste définissant le type d'objet et pouvant comporter une variable désignant une image de type mode point, une image graphique engendrée en l'absence de média
25 externe et une variable désignant un objet de type texte.

 En outre, une structure de données permet de définir la position du conteneur dans la scène selon le tableau 24.

 La structure de données précitée permet de définir
30 les positions de l'objet en fonction du type d'objet, les positions X et Y et dimensions de largeur maximale et de

hauteur maximale étant définies par rapport au numéro de pixel du moniteur d'affichage simulé MS, et donc en définitive du moniteur de la plate-forme ou terminal d'accès à l'application interactive considérée.

----- 5 ----- Enfin, chaque scène courante et le cas échéant
 chaque application comporte avantageusement une structure de données descriptive du couple événement/action, cette structure de données permettant la description d'une ligne de macro-commandes telle que décrite précédemment dans la
 10 description relativement à la scène courante ou le cas échéant à une application, selon le tableau 25.

Cette structure de données peut comporter avantageusement une variable d'identification de l'événement déclenchant, une variable de paramètre de l'événement,
 15 l'événement pouvant correspondre à l'un des événements tels que décrits au tableau 7 précédemment introduit dans la description par exemple, une variable de type booléen pour un événement de type continu, une valeur par défaut étant alors attribuée, une variable d'identification de
 20 l'action à déclencher, une première et une deuxième variable ou paramètres de description de l'action selon le type d'action considéré. En outre, pour chaque condition et afin d'assurer la gestion de la condition précitée, la structure de données comporte une variable de référence à
 25 une condition et une structure interne comprenant au moins une variable d'identification de la condition, une variable de gestion de la condition selon le premier chiffre et une variable de valeur pour exécution de l'action.

La règle de gestion du couple événement/action décrit par la structure de données précédente est réalisée
 30 par l'intermédiaire de la variable Act-Pass, et, à la va-

leur non vraie, l'action correspond à l'action Go-Scene du tableau 8, c'est-à-dire à l'initialisation d'une scène ou au changement de scène ainsi que mentionné précédemment dans la description.

5 Une description plus détaillée d'éléments particu-
lièrement remarquables du système de développement rapide d'applications interactives, objet de la présente invention, sera maintenant donnée en liaison avec la figure 2 et les figures suivantes.

10 D'une manière générale, on indique que ces éléments remarquables sont constitués par des éléments logiciels dont les pages écran, interface entre l'utilisateur du système de développement rapide d'applications interac-
15 présente des caractéristiques fonctionnelles spécifiques liées aux structures de données précédemment décrites dans la description.

 En particulier, dans le cas où le moniteur d'affichage, le module de commande et le système d'exploitation
20 de la plate-forme cible ou terminal d'accès sont ceux d'un récepteur de télévision, l'éditeur électronique E permet la simulation de l'écran d'affichage de ce récepteur de télévision et de l'interface de commande d'un module de télécommande universelle associé à ce récepteur de télévi-
25 sion, ainsi que représenté en figure 2.

 De structure classique, le boîtier à boutons comportant, ainsi que représenté sur la figure précitée, des boutons spécifiques de sélection de fonctions, désignée par SF, des boutons de sélection de valeurs numériques,
30 désignée par VN, des boutons de réglage de programmes RP, des boutons de navigation flèche en haut, flèche en bas, à

droite et à gauche, notés NV, et des boutons de codage de caractères alphanumériques AN, des boutons de menus et de radios par exemple, aux touches précédentes de nom correspondant sont associés les événements selon le tableau 26 de correspondance.

On comprend ainsi qu'à chaque touche correspondante dont le nom est indiqué au tableau et représenté au dessin est en fait associé un événement selon la désignation mnémotechnique de cet événement. Ainsi, pour la gestion du module de télécommande, constituée par une boîte à boutons, à chaque bouton de cette télécommande peut être associé un événement de la liste des événements introduits selon le tableau 7. A titre d'exemple, au bouton OK peut être associé l'événement KeyOK ou n'importe quel autre événement tel que AppliStart, TimerExpired par exemple. Cette gestion est réalisée grâce à un module logiciel qui permet d'effectuer des affectations croisées. Le module de télécommande permet ainsi de simuler tout module externe qui serait susceptible de produire une interaction.

Lorsque le module de commande et le système d'exploitation sont ceux d'une station de travail ou d'un ordinateur, l'éditeur électronique E permet la simulation de l'écran d'affichage de ce micro-ordinateur et de l'interface de commande d'un appareil périphérique d'entrée de ce micro-ordinateur, tel que par exemple le clavier. A une touche spécifique de cet appareil périphérique d'entrée est associé un événement provoquant la rupture d'équilibre de l'état du système d'exploitation. On comprend en particulier que les événements mentionnés au tableau 26 peuvent alors être associés à des touches spécifiques du clavier précité. De la même manière que dans le cas de la figure

2, lorsqu'un module de télécommande universelle associé à un récepteur de télévision est utilisé, la correspondance entre événements et touches du clavier est réalisée de manière logicielle. Une telle mise en correspondance ne sera pas décrite en détail car elle correspond à une opération de même nature que dans le cas de la figure 2.

Une description plus détaillée d'un ensemble d'outils logiciels constitutifs du système de développement rapide d'applications interactives, objet de la présente invention, sera maintenant donnée en liaison avec les figures 3a₁, 3a₂, 3a₃ à 3j₁, 3j₂.

La notion d'outil logiciel recouvre, conformément à l'objet de la présente invention, un ensemble de pages écran permettant de constituer de manière particulièrement souple les structures de données précédemment décrites dans la description. On comprend en particulier que les pages écran précitées constituent une interface pour l'utilisateur du système de développement rapide d'applications interactives, objet de la présente invention, afin de permettre à ce dernier de constituer les structures de données précédemment décrites dans les conditions de format et d'inter-opérabilité de ces structures de données nécessaires à la constitution d'applications interactives de type natif par exemple.

En référence à la figure 1a, on indique que l'éditeur électronique E, le module générateur d'objets virtuels GOV spécifique à l'application et le module de commande de mise en corrélation d'une pluralité de couples événement/action COR, comportent avantageusement un module logiciel pilote permettant notamment, à partir d'une fenêtre d'affichage d'une représentation du module de commande

simulé MCS, et le cas échéant d'appareils périphériques, de recevoir une variable "*EVENEMENT*" correspondante, ainsi qu'à partir d'une fenêtre d'affichage de scènes constitutives de l'application interactive en cours de développement, -- de -- provoquer, -- soit -- l'initialisation, -- soit -- la modification de la scène considérée, par l'intermédiaire d'une variable "*ACTION*", tel que mentionné précédemment dans la description.

On comprend toutefois que le module logiciel pilote comprend une pluralité d'outils logiciels, chaque outil logiciel étant sensiblement associé à au moins une fenêtre d'affichage permettant le développement des structures de données précédemment décrites dans la description par l'utilisateur du système de développement rapide d'applications interactives, objet de la présente invention.

Les fenêtres essentielles associées aux outils logiciels correspondants seront maintenant décrites en liaison avec les figures 3a₁ à 3j₂ ci-après.

Selon un premier aspect remarquable du système, objet de la présente invention, celui-ci comporte un outil de création de macro-instructions, lequel comporte au moins une page écran de mise en corrélation de couples événement/action, l'ensemble des pages écran essentielles associé à l'outil précité étant représenté en figures 3a₁, 3a₂ et 3a₃.

En référence aux figures précitées, on indique que les pages écran successives comprennent au moins une fenêtre d'affichage d'un ensemble d'événements, désigné par liste des Ev/Actions, une fenêtre d'affichage d'une action déclenchée associée à l'un des événements de cet ensemble d'événements et au moins une fenêtre de paramétrage de

l'action déclenchée, ces fenêtres étant notées Param 1, Param 2 sur les figures précitées. Un dispositif de pointage et de sélection, tel que par exemple le pointeur commandé par une souris, est prévu, ce dispositif étant géré
 ----- 5 ----- par le système d'exploitation du micro-ordinateur hôte -----
 constitutif du système de développement rapide, objet de la présente invention. Ainsi, la mise en corrélation de couples événement/action peut être effectuée par sélection d'un événement parmi l'ensemble des événements et de l'action
 10 tion par l'intermédiaire du pointage.

Sur les figures 3a₁, 3a₂ et 3a₃, on a représenté la sélection successive dans la fenêtre Liste des Ev/Actions d'un événement représenté par la désignation mnémotechnique des événements telle que donnée en figure 2 ou au tableau 7 précédemment mentionné dans la description.
 15 L'action peut être alors effectuée par sélection dans la fenêtre action déclenchée d'une action telle que l'action désignée par GoScène. Sur les figures 3a₁, 3a₂ et 3a₃, les différentes fenêtres sont reliées par des traits continus
 20 aux champs des macro-instructions ou lignes d'instructions constitutives des scripts. La figure 3a₁ représente ainsi la création de la ligne de script 1, la figure 3a₂ celle de la création de la ligne de script 2 et celle de la figure 3a₃ celle de la ligne de script 3 à partir des événements
 25 affichés dans le fenêtre de liste des Ev/Actions et sélection de chaque événement successivement, puis de l'action correspondante, ainsi que des paramètres dans les fenêtres Param 1 par exemple.

L'outil de création de macro-instructions décrit
 30 précédemment permet ainsi d'engendrer la structure de données correspondant aux macro-instructions et de gérer les

conditions de déclenchement des actions ainsi programmées. En mode édition, l'outil précité permet également de supprimer les conditions de déclenchement, d'effacer une ligne de macro-instructions par le bouton Supprimer. Cet

5 outil effectue ainsi une auto-corrélation sur les paramè-
 tres ainsi que sur les champs, le choix des paramètres dans les fenêtres de paramètres Param 1 et Param 2 n'étant proposé que par l'intermédiaire de menus que lorsqu'ils existent comme paramètres du champ actions.

10 De même, le type de la donnée introduite dans la fenêtre Param 1 impose un choix directif dans la fenêtre Param 2. A titre d'exemple, si la fenêtre Param 1 est chargée par une variable de type texte, la fenêtre Param 2 ne peut être que du type texte et l'éditeur ne propose

15 donc que des valeurs de même type.

Grâce à la mise en œuvre de l'outil précité, les erreurs de programmation sont ainsi minimisées, l'utilisateur-concepteur ne pouvant pas écrire une ligne de macro-commandes non exécutable. Enfin, les lignes de macro-

20 commandes sont hiérarchisées, la ligne 1 étant toujours exécutée en premier. Lorsque dans la fenêtre action déclenchée est choisie l'action GoScreen, une telle action provoque automatiquement un arrêt d'exécution de la macro-commande.

25 Le module pilote comporte également un outil logiciel de création d'objets virtuels dont les pages écran essentielles sont représentées en figures 3b₁ et 3b₂.

Ainsi que représenté sur les figures précitées, l'outil de création d'objets virtuels comporte au moins

30 une page écran de création d'objets comprenant au moins une fenêtre d'affichage d'une liste d'objets accessibles.

Cette liste d'objets système correspond à une liste des objets d'interface et est affichée selon une succession de ces objets et un champ de niveau, la succession des objets constituant un champ de désignation d'objets. En outre, la

5 page écran de création précitée comporte une fenêtre de
 choix de mode d'affichage/représentation de cet objet, un champ de type d'objet permettant de discriminer la nature de cet objet, une fenêtre de type d'objet permettant d'engendrer un champ de type d'objet permettant de discriminer

10 la nature de cet objet, ainsi qu'une fenêtre relative à un ensemble de paramètres fonction du média d'affichage/représentation de cet objet, cette fenêtre étant représentée sur les figures 3b₁ et 3b₂ par des fenêtres auxiliaires, des fenêtres de position X, Y, de hauteur et

15 de largeur du média considéré.

Sur les figures 3b₁ et 3b₂, on a représenté la définition concrète de deux objets, la correspondance entre les éléments introduits à partir de l'outil logiciel de création d'objets et la structure de données correspondante, et en particulier la structure de données

20 OC-(AllData) décrite dans le tableau 15 précédemment dans la description.

En outre, ainsi que représenté en figure 3c, le module pilote comporte un outil d'édition des médias comportant au moins une fenêtre page écran Média comprenant

25 au moins une fenêtre d'affichage d'une liste des plateformes d'exécution, une fenêtre d'affichage d'une liste de médias spécifiques susceptibles d'intervenir dans la composition d'une scène, chaque média étant défini par un nom

30 et un attribut, l'attribut alloué à un média donné permet-

tant de discriminer le caractère interne ou externe du média vis-à-vis de l'application considérée.

En référence à la figure 3c, on indique que l'outil d'édition de média permet ainsi l'ajout d'un média par le bouton Ajout ou la suppression de ce média lorsque le média n'est plus utilisé dans les objets ou dans les fonds constitutifs de l'application interactive considérée. Cet outil permet ainsi de caractériser le média, son nom et ses attributs et l'attribut chargement des médias peut être du type "ap" ou "sc" si le type du média est une image vidéo numérique de type Mpeg, le type "ap" indiquant que le média est inclus dans l'application et le type "sc" indiquant que le média est externe à l'application et sera capturé si besoin lors de la présentation d'une scène.

En référence à la figure 3d, on indique que le module pilote peut également comporter un outil d'édition des fonds, cet outil permettant d'ajouter et de supprimer si nécessaire des fonds et d'affecter les données selon la structure de données de fonds telle que décrite en liaison avec les tableaux 11 et 12 précédemment mentionnés dans la description.

Ainsi que représenté en figure 3e, le module pilote comporte de manière avantageuse un outil d'édition de phrases sous forme de fichier texte, chaque phrase comportant un numéro d'identification permettant d'affecter cette phrase, soit au niveau de l'application interactive en cours de développement, soit au niveau d'une ou plusieurs scènes constitutives de cette application. On comprend en particulier que l'outil logiciel précité permet de saisir essentiellement du texte, désigné par phrase, des données correspondantes de type phrase étant affectées

au niveau application ou au niveau de chaque scène constitutive de celle-ci.

De même, en référence à la figure 3f, on indique qu'un outil logiciel d'édition de variables peut être prévu, cet outil permettant de créer ou de supprimer si besoin des variables de contenu texte ou numérique, ces variables étant structurées selon la structure de données VA_(AllData) introduite précédemment dans la description en liaison avec le tableau 14.

10 Selon un mode de réalisation préférentiel du système de développement d'applications interactives, objet de la présente invention, on indique que le module pilote peut également comporter un outil logiciel de création de scène comportant au moins une fenêtre d'affichage d'une
15 liste de scènes types, ainsi que représenté en figure 3g, chaque scène type étant repérée par une désignation spécifique. En outre, une fenêtre de définition de paramètres de la scène choisie peut être prévue, paramètres tels que numéro d'identification de scène, désignés par Scène ID,
20 palette de couleurs, choix du fonds à l'initialisation de la scène, désigné par Background à l'openScène, Position de départ du curseur, et une fenêtre de choix d'Options pour le curseur considéré.

L'outil logiciel de création de scènes permet de
25 créer une nouvelle scène, respectivement de dupliquer ou de supprimer une scène existante. Lors de la duplication d'une scène, toutes les données relatives à la scène sont copiées au niveau de la scène dupliquée.

En ce qui concerne les options telles que les options du curseur précédemment mentionnées en liaison avec
30 la figure 3g, on indique que le module pilote comporte

également un outil logiciel de définition d'options dans une scène dont les pages écran sont représentées aux figures 3h₁ et 3h₂.

5 D'une manière générale, on indique que l'outil précité permet, à partir d'une liste d'options définies pour la scène considérée, en particulier les options du curseur, d'attribuer une option déterminée parmi la liste d'options et de naviguer et choisir une option auxiliaire autour de l'option attribuée à partir d'un ensemble de
10 boutons de commande du module de commande associé au moniteur d'affichage. On comprend en particulier, en référence à la figure 2 par exemple, que les boutons de commande concernés peuvent avantageusement être des boutons de déplacement vertical, vers le haut, vers le bas, vers la
15 droite, vers la gauche appartenant au groupe des boutons de navigation NV, tel que représenté en figure 2 et au tableau 26 précédemment introduit dans la description.

On comprend en particulier que grâce à l'outil logiciel précité, il est ainsi possible d'attribuer une option de départ à la constitution d'une scène, puis ensuite
20 de naviguer par l'intermédiaire des boutons préconfigurés, flèches, ou de choisir une autre option par le choix d'un bouton du clavier ou d'un événement couplé à l'action de désignation d'une option DesignOpt.

25 L'outil logiciel précité peut également inclure de manière avantageuse l'exécution directe de l'action, offerte parmi toutes les actions offertes, couplée à l'option désignée si l'événement Keyok est déclenché. Une analogie de fonctionnement avec le double cliquage d'un
30 bouton de la souris d'un micro-ordinateur peut être mise en évidence.

Deux exemples illustratifs spécifiques seront donnés ci-après.

Exemple 1

Création d'une animation d'objets, virtuels. Chaque objet dispose d'un attribut d'affichage de type option et d'un numéro d'option de 1 à 10. La sélection séquentielle de chaque option permet donc de faire défiler les objets les uns derrière les autres et de restituer ainsi un kaléidoscope d'objets.

10 Exemple 2

Création d'un curseur à l'écran qui matérialise une désignation et une sélection par la boîte à boutons, par utilisation de quatre flèches.

En conséquence, l'option attribuée constitue en fait un paramètre d'affichage de l'objet virtuel ainsi constitué, ce qui permet de conférer aux applications interactives développées grâce au système, objet de la présente invention, une très grande souplesse et une très grande versatilité d'utilisation.

20 De même, afin de caractériser chaque application interactive développée grâce au système, objet de la présente invention, cette caractérisation visant aussi bien des caractères spécifiques au contenu de l'application, au numéro de version de l'application et ses évolutions ainsi qu'à des droits, notamment droits d'auteur relatif à cette application, le pilote peut comporter avantageusement un outil d'émission des attributs d'une application déterminée, ces attributs comportant au moins un champ commentaire descriptif de l'application, un champ de désignation de l'auteur de l'application, un champ de nom de l'application, un champ de numéro de version de l'application et

un champ de palette de couleurs de l'application considérée, ainsi que représenté en figure 3i.

Selon un aspect particulièrement remarquable du système de développement d'applications interactives, objet de la présente invention, pour chacune des plateformes d'exécution d'application, celui-ci comprend un ensemble de tables de données communes au couple plate-forme d'exécution/application et un outil de consultation des tables communes permettant de consulter les tables communes précitées, manipulées dans l'éditeur E, pour les plateformes d'accès définies.

L'ensemble des données communes comporte au moins :

- une table des données d'applications relatives à la plate-forme considérée ;
- une table des données caractéristiques de chaque plate-forme ;
- une table des événements relatifs à la plate-forme d'accès considérée ;
- une table des actions d'applications relatives à cette même plate-forme ;
- une table des fontes disponibles pour la plate-forme d'accès considérée.

La table des données communes peut être définie pour chaque application, à l'éditeur E étant associé un fichier texte particulier qui contient un descriptif des données manipulées.

Un tel fichier texte, dénommé ADR_Data, peut être édité par un simple éditeur de texte et permet d'engendrer des données externes. La modification de ces données entraîne, par l'intermédiaire de l'éditeur E, une mise à

jour de ces données dans l'application en cours ouverte. Lors de l'ouverture de l'application, une recopie de ces données est effectuée de l'éditeur dans l'application.

Un exemple de contenu du fichier ADR_Data est donné ci-après dans le tableau 27.

Dans ce mode de mise en œuvre de la structure de données représentée au tableau 27, les trois premières lignes définissent des données manipulées à travers des API, pour *Application Protocol Interface*. Les deux premières données sont intrinsèques au pilote ADRTVnum, TheKeyNum permettant d'identifier une touche du clavier et TheCurOpt permettant d'identifier le numéro d'option courant. TheCurTime à travers une API spécifique pour chaque plateforme d'accès permet d'identifier l'heure courante. Des extensions peuvent bien sûr être prévues telles que des données définies dans le système DVB ou des données liées à un système de contrôle d'accès par l'intermédiaire d'une carte à microprocesseur ou d'un modem.

Ces données manipulées à travers le module pilote sont statiques car invariantes, mais restituées d'une façon dynamique sur une plateforme d'accès de type terminal récepteur de télévision par exemple.

Un exemple de création d'une structure de données correspondante telle que représentée au tableau 6 précédemment dans la description pour les données désignées par la structure de données DO_AllData est représentée en figures 3j₁ et 3j₂ ci-après, la figure 3j₁ permettant la mise en correspondance de la table des données et des listes de plates-formes à partir des valeurs d'identification de données Data_ID, origine de la mise à jour des données, type de la donnée et valeur actuelle, et de paramètres re-

latifs à la plate-forme utilisée, la plate-forme MACINTOSH dans le cas de la figure 3j₁ et des paramètres relatifs à cette plate-forme, et la figure 3j₂ représentant les fenêtres relatives aux pages écran, aux actions, aux événements et aux fontes, ainsi que mentionné précédemment dans la description.

Alors que l'ensemble des structures de données et des macro-instructions développé ainsi que décrit précédemment grâce à la mise en œuvre du système, objet de la présente invention, constitue une application interactive native, c'est-à-dire une application interactive dont l'ensemble des couples événement/action mis en corrélation et des scènes successives constitutives de cette application, ont été définis de manière générique pour tout type de terminal d'accès à l'application interactive considérée, bien entendu, le système, objet de l'invention, comporte en outre un module de traduction de l'application interactive native précitée en une application interactive dédiée à un terminal d'accès de type déterminé.

On comprend en particulier que l'application native étant décrite ainsi que mentionné précédemment, il est alors nécessaire d'engendrer une nouvelle description de l'application permettant la représentation de cette dernière dans le format du langage du pilote de la plate-forme cible ou terminal d'accès.

Ainsi que mentionné précédemment dans la description, le module de traduction de l'application interactive native peut consister en différents sous-modules de programmes permettant la traduction de cette application native en une application interactive dédiée à un terminal

de type *OPEN TV*, MAC-OS ou document HTML, ou tout autre système d'exploitation.

Une description plus détaillée de l'organisation générale de la création d'une application interactive native puis de la traduction de cette dernière en une application dédiée de type *OPEN TV* à partir des structures de données précédemment décrites sera maintenant donnée en liaison avec les figures 4a et 4b.

Sur la figure 4a, on a représenté l'organisation générale de la création de l'application à partir des fichiers d'origine et des fichiers courants auxiliaires représentés par des ovales à partir de modules d'exécution de programmes représentés par des rectangles.

Dans le cas où l'application comporte des graphiques et des images naturelles, les médias représentés par des fichiers A et B et leur transformation respective sont nécessaires pour chaque pilote considéré. Les transformations de formats à partir d'images naturelles sont des opérations effectuées par des convertisseurs de format externe à l'Atelier ADRTVNum. Un convertisseur de format 200A est prévu pour transformer l'image naturelle représentée par le fichier A en un fichier A' pour une représentation image à travers le pilote MacOS. Un convertisseur de format appelé codeur image Mpg2, appelé ici module de programme 203, est nécessaire pour transformer l'image naturelle représentée par le fichier A en fichier N pour une représentation image à travers le pilote Opentv.

Dans le cadre de la transformation d'images dites graphiques, un convertisseur de format 200B est nécessaire pour la transformation d'une image graphique B en un format représenté par le fichier D pour le pilote MacOS et un

fichier C pour le pilote Opentv. Un module logiciel 202 permet, à partir du fichier C, d'engendrer un fichier F descriptif de l'objet constitué par l'image précitée, taille exacte du média dans le format Opentv après transformation finale, table des couleurs, et un fichier G par codage de l'image d'origine par un codage spécifique Opentv. Les fichiers F et G portent les noms pixName.txt et pixName.rc respectivement.

Le fichier H, appelé Datamake.txt, constitue le fichier modèle permettant de générer un fichier de diffusion appelé Q en Opentv.

Lorsque la complétude est constatée, c'est-à-dire lorsque la présence des différents fichiers est nécessaire au bon fonctionnement de la génération, a été vérifiée, un module de programme spécifique permet la création des fichiers descriptifs de l'application pour un système d'exploitation déterminé. Les fichiers descriptifs portent la référence I, J, K, L sur la figure 4a et les modules de programme précités, portent la référence 206 pour une application Opentv, le module 208 permet une génération pour une application document en HTML et le module 207 pour un format quelconque.

A partir des fichiers I, J, K, L précités et de fichiers auxiliaires tels que fontes utilisées, fichiers M, les fichiers images MPEG N établis à partir du module 203, du fichier pilote O et du fichier P de classe description générique pour toutes les applications, un module de programme 209 permet d'assurer une compilation en un fichier U constituant une application interactive dédiée prête à être distribuée. Un module de programme 210 permet l'exécution de l'application représentée par le fichier Q.

Les fichiers images MPEG N peuvent se situer au même niveau que le fichier G, de manière que Checkoptv permette de vérifier leur présence.

Dans le cas d'une application dédiée au système *OPEN TV*, un outil de génération spécifique est prévu au niveau de l'éditeur E dont une page écran est représentée en figure 4b. Cet outil permet en particulier d'engendrer des simplifications de mise en œuvre propres aux terminaux d'accès *OPEN TV* tels que :

- 10 - message trace : lorsque la case correspondance est cochée, ceci permet d'activer une fonction de trace dans le logiciel d'application avec affichage du descripteur, champ commentaire descriptif d'une application, champ auteur de l'application, taille mémoire restante,
- 15 taille mémoire continue restante par exemple ;
- mode vidéo : suivant les versions de réalisation de la plate-forme d'accès *OPEN TV*, la surface d'incrustation est plus ou moins restreinte ;
- effacer la scène : permet d'éviter les effets visuels
- 20 entre l'effacement d'une scène et la construction de la nouvelle scène ;
- délai : permet de gérer les temps inter-modules et influant sur la construction du fichier de construction du flux de données, le fichier J.

25 Une description plus détaillée d'un gestionnaire de développement d'applications interactives à diffusion cyclique, accessibles notamment sur un canal TV à partir d'un terminal d'accès muni d'un décodeur TV, sera maintenant donnée en liaison avec les figures 5a et 5b.

30 Lorsque la capacité mémoire des décodeurs, notamment des décodeurs TV, est limitée, il est nécessaire de

mettre en œuvre un processus de prédiction ayant pour objet d'agir sur l'acquisition des médias des données programme propres à la construction des futures scènes d'une application donnée.

5 On rappelle en premier lieu que les objets virtuels sont des conteneurs qui contiennent des contenus médias ou multimédias, c'est-à-dire en fait des objets distincts.

10 Ainsi que représenté sur la figure 5a, le système objet de la présente invention comporte au moins dans ce but un module logiciel M_1 gestionnaire d'une pluralité de scènes successives constitutives de cette application, selon un niveau de profondeur de chaque scène vis-à-vis de la scène courante. Le module logiciel gestionnaire M_1 permet d'analyser et de discriminer sur un nombre déterminé
15 de scènes successives voisines de la scène courante des objets virtuels communs et les médias communs associés à ces objets de ces scènes successives, des objets virtuels distincts et des médias distincts associés à ces objets de
20 ces scènes successives.

 Selon un mode de réalisation non limitatif, on indique que les objets virtuels et les médias correspondants pour chaque scène considérée peuvent être constitués selon une liste informatique sur laquelle les opérations du module logiciel M_1 peuvent être effectuées. Ainsi, la liste
25 des objets virtuels et des médias relative à la scène courante est notée L_c et la liste des objets virtuels et des médias relative à chaque scène successive de rang 1 à p voisine de la scène courante est notée L_{c+1}, \dots, L_{c+p} .

30 Par des opérations informatiques analogues à des opérations de troncature et d'union d'ensembles appliquées

aux listes précitées, le module logiciel gestionnaire M_1 permet de délivrer des listes désignant les objets virtuels communs et les médias communs associés à ces objets de ces scènes successives, ces listes étant notées

5 $L_{1c}, L_{2c}, \dots, L_{pc}$ pour les listes communes. A titre d'exemple non limitatif, les listes communes précitées peuvent être déterminées par l'opération de troncature entre la liste courante L_c et la liste suivante commune L_{c+1} et actualisation successive. Une pluralité de listes courantes L_{c+1} à

10 L_{c+r} avec $r > p$ peut être prise en considération afin d'optimiser les calculs. De même, le module logiciel gestionnaire M_1 permet de calculer les listes distinctes d'objets virtuels et des médias distincts, notées $L_{1d}, L_{2d}, \dots, L_{pd}$ par une opération de disjonction entre la liste courante L_c et

15 les listes successives $L_{c+1}, L_{c+2}, \dots, L_{c+p}$. De la même manière que dans le cas des listes communes, des listes distinctes peuvent être réalisées pour la liste courante à partir d'une pluralité r de listes successives L_{c+1} à L_{c+r} , $r < p$.

Ainsi que représenté en outre en figure 5a, le

20 système comporte également un module logiciel d'actualisation M_2 des objets virtuels et médias communs et des objets virtuels et médias distincts pour au moins une scène successive constitutive de la pluralité de ces scènes successives par rapport à la scène courante. Ainsi que représenté sur la figure précitée, le module logiciel

25 d'actualisation M_2 reçoit les listes communes et les listes distinctes précédemment mentionnées.

On comprend ainsi qu'à partir de ces dernières, le module logiciel d'actualisation M_2 permet de calculer

30 chaque liste de mise à jour, notée L_{mi} , à partir de chaque

liste commune successive L_{ic} et de chaque liste différence successive L_{id} .

Les listes précitées vérifient la relation symbolique :

$$5 \quad L_{mi} \leftrightarrow (L_{ic}, L_{id}).$$

Un exemple spécifique de mise en œuvre du module logiciel représenté en figure 5a sera maintenant donné en liaison avec la figure 5b.

D'une manière générale, on indique que les données
 10 multimédias ou les médias considérés sont chargés dans des
 conteneurs propres à la diffusion. Ces conteneurs spécifiques
 sont désignés par modules conteneurs. En outre, le
 logiciel de l'application interactive considérée est également
 emballé dans un ou plusieurs modules conteneurs.
 15 Le module conteneur nécessaire à la gestion de la première
 scène de l'application multimédia considérée est appelé
 module principal. Ce module doit être capturé en premier
 par le terminal afin de permettre la représentation de la
 première scène de l'application interactive. Un module
 20 conteneur peut contenir plusieurs médias ou multimédias
 mais strictement de même type. La taille du module conteneur
 de chaque type est fixée par l'éditeur, c'est-à-dire
 par l'outil ou système de développement rapide d'applications
 interactives, objet de la présente invention.

25 Le degré de profondeur d'une scène de l'application
 désigne le niveau de cette scène par rapport à la
 scène courante.

Par convention, dans l'exemple donné en relation
 avec la figure 5b, on indique que le niveau zéro désigne
 30 la scène courante présente à l'écran, le niveau 1 la pro-

chaîne scène contiguë à la scène courante, les différents niveaux successifs 2 à p étant ainsi définis.

Une analyse fonctionnelle de l'application et la mise en œuvre d'un prédicat relatif à la nécessité de mémoriser ou non chaque média et chaque objet virtuel permet alors d'optimiser la mémorisation et les capacités de mémorisation nécessaires à la conduite de la représentation des scènes.

Les paramètres nécessaires à l'analyse sont les suivants :

- taille mémoire allouée à une application interactive téléchargée dans un terminal ;
- structure de l'application interactive et en particulier arbre et contenu de chaque scène ;
- taille mémoire de chaque conteneur de diffusion par type de média ;
- nombre de niveaux d'analyse, c'est-à-dire r, pour chaque scène.

Le processus, et en particulier le module logiciel gestionnaire M_1 , analyse la structure arborescente de l'application et constitue pour chaque scène successive un arbre de diffusion contenant toutes les scènes de niveau 0 à p.

- A chaque scène de niveau 0 sont constituées les listes des différents chemins correspondant aux futures différentes scènes possibles de sortie, c'est-à-dire de niveau 1.

- A chaque scène de niveau 0, une analyse des conteneurs est effectuée et à chaque conteneur est affecté une variable de mémorisation de la forme :

■ 0 : acquisition lorsque la scène est ouverte ;

- 1 : acquisition lorsque l'application interactive est téléchargée ;
- 2 : gestion par l'automate.

L'affectation 0 ou 1 est proposée par le système de développement, objet de la présente invention, au concepteur de l'application, par l'intermédiaire de l'éditeur E et de l'écran interactif.

- A chaque scène de niveau 2 à p, une analyse des conteneurs de diffusion est effectuée et, à chacun de ces derniers, est affectée une variable de mémorisation de la forme :

- 1 : le conteneur de diffusion doit être acquis lorsque la scène en cours est représentée ;
- 0 : le conteneur ne doit pas être acquis lorsque la scène en cours est représentée.

L'affectation des valeurs 0 ou 1 précédente est effectuée de manière globale, pour chaque scène de niveau compris entre 2 et p, ou pour un nombre de scènes r, ainsi que mentionné précédemment dans la description.

Ainsi, une première passe d'analyse et de stratégie d'acquisition des conteneurs de diffusion est effectuée. Lors de cette passe, le module logiciel gestionnaire M_1 constitue ainsi la liste des conteneurs nécessaires à chaque scène et une seconde passe permet de compléter cette liste et de déterminer les modules de diffusion suivant la stratégie d'acquisition afin d'optimiser l'occupation mémoire d'un module. La taille mémoire occupée à chaque scène par l'ensemble des modules est ainsi connue.

En fonction de la taille mémoire disponible dans le terminal d'accès, le concepteur de l'application peut ainsi affiner sa stratégie et effectuer un compromis entre

la vitesse d'affichage d'une scène avec tous ses conte-
neurs de diffusion et la taille mémoire allouée aux conte-
neurs multimédias.

La taille mémoire de l'application interactive né-
cessaire au téléchargement est ainsi optimisée : plus
cette taille est faible et moins le temps d'attente d'ap-
parition de la première scène de l'apparition est long.
L'analyse précédente peut également être adaptée pour con-
naître la ressource nécessaire en débit pour une applica-
tion interactive considérée.

Selon un aspect remarquable du système objet de la
présente invention, à chaque changement de scène, le pi-
lote du terminal d'accès vérifie, par l'intermédiaire du
module logiciel d'actualisation M_2 , si les conteneurs de
diffusion à acquérir ne sont pas déjà en mémoire. La place
mémoire des mots nuls non nécessaires est ainsi libérée.

La figure 5b représente les relations successives
entre les scènes d'une application interactive donnée à
titre d'exemple non limitatif et composée de sept scènes.

Chaque scène est réputée contenir trois médias de
type différent, les médias, pour chaque scène, étant défi-
nis par un indice de média et un indice de scène, la nota-
tion média jk désignant ainsi le média de rang j utilisé
dans la scène de rang k .

Chaque scène sur la figure 5b est réputée compor-
ter trois médias de type différent. A chaque scène de ni-
veau 2 à p , l'acquisition n'est pas demandée pour chaque
média.

Le tableau résultant de l'analyse de l'application
interactive correspondant à l'exemple donné en figure 5b
est le tableau n° 28 en annexes.

A titre de simplification, on a volontairement réduit le tableau 28 à quatre niveaux. Enfin, dans le tableau 28, l'indice de scène k a été supprimé mais la valeur de la variable binaire, acquisition ou non du conteneur, a été

5 mentionnée.

A titre d'exemple non limitatif simplifié, chaque média étant de même type et étant réputé nécessiter un espace mémoire de 30 k octets, la taille mémoire requise pour gérer, dans l'exemple de la figure 5b, un arbre de

10 niveau 1 est $12 \times 30 = 360$ k octets.

ANNEXES**TABLEAU 1**

(100,101)→102 ; (105,102,104)→106 ; 103→104 ;
(107)→108 ; (109,110,106,108,111,116,117,113)→ 114 ;
(116,111)→115 ; (111)→116 ; (116)→117 ;
(117,116,115,111,108,104,102,112,118)→113 ;
(114)→119.

TABEAU 2

PL		Table des plates-formes cibles			
PL_MaxTextSize		255	Nbr max de char pour les textes (var/Don et phrases)	fld PL_MaxText	
Pour chaque PL					
PL_Ident			Identifiant de la plate-forme	fld PL_AllData	1 ligne
PL_Name			Nom de la plate-forme		item 1
PL_Nbr_Coul		num	Nombre de couleurs max de la palette (attention : OpenTV : 15)		item 2
PL_RamMedia Max		koctet	Mémoire maxi (pour chargement média)		item 3
PL-FileMedia Size		koctets	Espace dispo pour média (disque/Module)		item 4
PL_FileMedia Max		num	Nbr max de média (par disque/Module)		item 5
PL_EndOfLine			Caractère de fin de ligne pour les textes		item 6
PL_ScreenRect OK		x1,y1,x2,y2	Zone de sécurité de la plate-forme (ex. SAGEM/mode2)		item 7
Cette table peut être étendue à d'autres paramètres tels :					
Opacité de la couleur de la transparence		0 à 255	Efficacité de la transparence		items 8 à 11

TABLEAU 3

AC	Table des actions				
	Pour chaque action			IdAC_AllData	1 ligne
	AC_Ident		Identifiant de l'action		item 1
	AC_Type	<AC_Type_list>	Type de l'action (selon ce sur quoi l'action porte)		item 2
	AC_Name		Nom de l'action		item 3
	AC_NbPar	num	Nombre de paramètres		item 4
	Pour chaque PL		⇒ pour les types dépendants des PL	card PL_j	j
		AC_Dispo	Vraie si action disponible pour cette plate-forme		item 2
		AC_Data	Si dispo : description de l'action pour la plate-forme		item 3
			Ex : nom de la fonction ou de la procédure à appeler		

TABLERAU 4

FO	Table des fontes		--v1 de l'éditeur : Hel 18, 24 et 36 seulement	
Pour chaque fonte			fid FO-AllData	1 ligne
	FO_Ident		Identifiant de la fonte	item 1
	FO_Name		Nom de la fonte	item 2
	Pour chaque PL		card PL-j	j
		FO_Ref	Référence de la fonte pour la plate-forme cible. Ex.	items 2 et +
			• MacOS : fo_ident,nom de fonte + taille	
			• Internet : nom de fonte + taille	
			• OpenTV : fo_ident,otv_ident	

TABIEAU 5

EV	Table des événements			(5 mini : Up,Down,Right,Left,OK⇒ type <Remote>)		
Pour chaque événement					fIdEV_AllData	1 ligne
	EV_Ident		Identifiant de l'événement			item 1
	EV_Type	<EV_Type_list>	Type de l'événement (selon ce qui génère l'événement)			item 2
	EV_Name		Nom de l'événement			item 3
	Pour chaque paramètre			(optionnel)		k
		EV_Param	Paramètres de l'événement (selon le type)			items 3+k
	Pour chaque PL _i				card PL _j	j
		EV_Dispo	bool	Vraie si événement disponible pour cette plate-forme		item 2
		EV_Data	si dispo	Si dispo : description de l'événement pour la plate-forme		items 3 et +
				Ex. nom du message reçu dans le code		

TABLERAU 6

DO	Table des données			
Pour chaque donnée			fId DO_AllData	1 ligne
	DO_Ident	1 à 3	Identifiant de la donnée – Data_ID=100+DO_Ident	item 1
	DO_Origine	<EV_Type_List>	Origine de la donnée : les READ ONLY seulement (pas les <appli>)	item 2
	DO_Type	<VA_Type_List>	Type de la donnée⇒ <num>ou <txt>	item 3
	DO_Name	txt	Nom de la donnée	item 4
	PH_Line	Lignes de texte (sans virgule) ou nombre entier		item 5
	Pour chaque PL		card PL_j	j
	DO_Dispo	bool	Vraie si donnée disponible pour cette plate-forme	item 2
	DO_Data	si dispo	Si dispo : description de la donnée pour la plate-forme	item 3
			Ex. nom de la fonction API qui retourne la donnée ..	

TABLEAU 7

Événement	Identification	Type	Liste Paramètres	Commentaires : Événements provoqués par :
Keynav	1	Remote	Up, down, right, left	Appui sur une des touches de navigation du clavier ou de la télécommande
KeyNum	2	Remote	0,1,2,3,4,5, 6,7,8,9,x	Appui sur une des touches du pavé numérique du clavier ou de la télécommande
KeyOK	3	Remote		Appui sur la touche de validation
KeyHome	4	Remote		Appui sur la touche Menu
KeyBack	5	Remote		Appui sur la touche Retour
AppliStart	6	Appli		L'application a été lancée et provoque ce message
SceneOpen	7	Appli		A chaque ouverture d'un événement est généré
TimerExpired	8	Appli	1 à 20	Lorsqu'un timer a fini son temps, cet événement est généré
SceneClosed	9	Appli		A la fermeture d'une scène, cet événement est provoqué
event1	10	??		Pré-câblage d'un événement à définir
event2	11	??		Pré-câblage d'un événement à définir
event3	12	??		Pré-câblage d'un événement à définir
event4	13	??		Pré-câblage d'un événement à définir
event5	14	??		Pré-câblage d'un événement à définir
MpegAff	15	Appli		Lorsqu'une image dans le flux diffusé est capturée et mémorisée, ce message est généré
InsertCard	16	Appli		Carte à puce introduite dans le lecteur
RemoveCard	17	Appli	0	Carte arrachée
Receptmodem	18	Ext	0	Demande de connexion

TABLEAU 8

Action	Identification	Type de l'action	Nombre de Paramètres	Commentaires:
Setbg	1	Med (media)	1	Affiche un background
PlayS	2	Med	1	Joue un son synthétique
PlayA	3	Med	1	Joue un son diffusé
PutInto	4	Var (variable)	2	Affecte une donnée à une variable
AddTo	5	Var	2	Additionne 2 nombres ou concatène 2 chaînes de caractères
GoScene	6	Nav (navigation)	1	Change de scène
GoBack	7	Nav	0	Retourne à la scène précédente
DesignOpt	8	Nav	1	Change d'option
LaunchApp	9	Nav	0	Lance l'application
QuitApp	10	Nav	0	Quitte l'application
ShowObj	11	Nav	1	Affiche l'objet
HideObj	12	Obj	1	Cache l'objet
SetTimer	13	Dec (décodeur)	2	Arme un timer à une certaine valeur en millisecondes
Selectserv	14	Dec	1	Sélection du service TV (provoque un zapping)
StopTimer	15	Dec	2	Arrête le ou tous les timers
Mute	16	Dec	0	Coupe le son
Unmute	17	Dec	0	Rétablit le son
Gl_actn	18	??	2	Action pré-câblée mais qui est définie par l'utilisateur
ReadUA	19	Dec	1	Lire le paramètre Ua d'une carte à puce
Phone	20	Dec	1	Numéroter et établir une communication par modem
Nothing	21	Dec	0	Ne fait rien – valeur de défaut
Multo				Multiplication entre 2 entiers numériques
Divto				Division entre 2 entiers numériques

TABIEAU 9

AP	Attributs application				AP	ReadAttAP
AP_Desc		Description de l'application	fId AP_Desc	all		GL_AP-Desc
AP_Author		Auteur, mention de ©	fId AP_Author	all		GL-AP-Author
AP_Name		Nom de l'application	fId AP_OneData	line 1		GL-AP-OneData
AP_Version		Version		line 2		
AP_Palette	<PA_Ident>	Palette de couleurs par défaut (si pas défini en scène)		line 3		
AP-StartSCRead	<Data_ID>	Donnée contenant le SC_Ident de la scène de départ (sinon : HOME)		line 4		Si rien :- 1
AP_CouTrans	<CO-Ident>	Couleur de transparence de l'appli		line 5		
Pour chaque PL		Indiquer quelles plates-formes on va générer		J		
AP-GenPL	Bool	Vraie si l'application devra être générée pour cette PL	line 6	item j		

TABIEAU 10

PA	Liste des palettes utilisées par l'application		PA	ReadListPal
Pour chaque palette gérée		fIdPA_AllData	1 ligne	GL_PA_AllData
	PA_Ident	Identifiant de la palette	item 1	
	PA_Name	Nom de la palette (pour faciliter l'édition)	item 2	
	Pour chaque couleur possible	15 couleurs (pas de coul transp vidéo) pour OpenTV	k	
	Pal-Ref	<CO_Ident>	Identifiant de la couleur associée à position dans palette	item k + 2

TABIEAU 11

BG	Liste des backgrounds utilisés par l'application		BG	ReadListBg
Pour chaque background géré		fId BG_AllData	1 ligne	GL_BG_Alldata
	BG_Ident	Identifiant du background	item 1	
	BG_Type	<Bgn_Type_List>	Type de background	item 2
	BG_Par		Valeur du background (selon le type)	item 3
	BG_Wait	<BG_Ident>, aucun	Bg de type <col>, obligatoire si Bgn_Type=<mpeg>SC	item 4
	BG_Name	Txt	Nom du background (pour faciliter l'édition)	

TABIEAU 12

Bgu_Type_list	Type d'un background							
	<col>	Couleur unie						
	<vidéo>	Transparence au signal vidéo						
	<mpeg>	Image MPEG en arrière-plan						
	<pict>	Image bitmap (recouvre en "tile")						

TABLEAU 13

ME	Liste des médias de l'application						
Pour chaque média géré				Id ME_AllData	1 ligne	ME	ReadListMedia
ME-Ident		Identifiant du média			item 1		GL_ME_AllData
ME_Type	<ME_Type_List>	Type du média			item 2		GL_ME_AllName
ME_Load	<"AP" ou "SC">	Mis à jour par la génération ?			item 3		
ME_GetorExit	Bool	Si vraie et pas de média : quitte => lorsque ME_Load="AP" (= false si SC)			item 4		GetOneMedia (ME_Ident)
ME_Usage	<int> ou <ext>	Si externe : éditable en mode "exploitation"			item 5		GL_ME_OneData
Pour chaque PL					1		GL_ME_OneName
ME_RefIdent		Référence du média pour une plate-forme			item 5 + j		
Structure interne : 1 ligne par ME_RefIdent			Id_MEAllRef	1 ligne			ReadListMeRef
ME_RefIdent		Identifiant de la référence			item 1		GL_ME_AllRef
ME_RefPL		Rappel: plate-forme de cette référence (PL-Ident)			item 2		
ME-Format		Format du média référencé (pct,gif,mpg,bmp..)			item 3		GetMeRef(ME_RefIdent)
ME_FileSize		Taille physique du média (pic/Module..)			item 4		GL_ME_OneRef
ME_RamSize		Taille mémoire (pour pré-chargement)			item 5		
ME-Ref		Référence du média pour la plate-forme			item 6		
		MacOS : chemin complet sur le disque (pathname)					
		Internet : URL					
		OpenTV : nom fichier contenant le média (dans rep DataSunToMac relatif appli et sans suffixe)					
ME_Name		Nom du média	Id ME_AllName	1 ligne			

TABLERAU 14

VA	Liste des variables de l'application			VA	ReadListVar
Pour chaque variable		Au moins une variable de type texte	Fid VA_AllData	1 ligne	GL_VA_AllData
VA_Ident	1 à 99	Identifiant de la variable – Data_ID = 200 + VA_Ident		item 1	
VA_Origine	<appli>	Origine de la donnée : READ/WRITE seulement => <appli>		item 2	
VA_Type	<VA_Type_List>	Type de la variable => <num> ou <tex>		item 3	
VA_Name	txt	Nom de la variable		item 4	
VA_Line		Lignes de texte (sans virgule) ou nombre entier		item 5	

VA-Type_List	Types d'une variable d'application				
	<txt>	Peut contenir toute chaîne de caractères			
	<num>	Nombre entier			

TABIEAU 15

OC		Liste des objets d'interface communs à toutes les scènes			
Pour chaque objet commun			fldOC_Alldata	1 ligne	OC
	OC_Desc	Description de l'objet commun			
	Descr_Objet	Structure unitaire de description d'un objet	(voir plus loin)	items	GL_OC_AllData

TABIEAU 16

AA		Lignes Event/Action attachées à l'application			
Pour chaque action "appli"			fldAA_Alldata	1 ligne	AA
	AA_Num	Numéro d'ordre de l'action		<i>num ligne</i>	
	AA_Desc	Description de l'action "appli"			GL_AA_Alldata
	Descr_Action	Structure unitaire de description d'une action	(voir plus loin)	items	GL_AA_Allref

TABIEAU 17

SC	Attributs propres à une scène		fid SC_OneData	SC	ReadAttSc
SC_Ident		Identifiant de la scène (toujours au moins une scène : HOME)	cd id	line 1	GL_SC_OneData
SC_Name		Nom de la scène	cd name	line 2	
SC_Palette	<PA_Ident>	Palette de couleurs (si vide : prend celle de l'appli)		line 3	
SC_BgFirst	<BG_Ident>	Identifiant du background pour l'open scène		line 4	
OP_Dep	OP-Num	Position de départ du curseur (-1 = pas de curseur)		line 5	
OP_Keep	bool	Vraie : repart du dernier opt de la scène		line 6	

TABIEAU 18

OS	Liste des objets d'interface de la scène		OS	ReadListObj (sc_Ident)
Pour chaque objet		fid OC_AllData	1 ligne	
OS_Desc		Description de l'objet		
	Descr_Objct	Structure unitaire de description d'un objet	(voir plus loin)	items
				GL_OS_AllData

TABLEAU 19

AS	Liste des Evénements/actions attachés à la scène	fId AA_AllData	1 ligne	AS	ReadListObj (sc Ident)
	Pour chaque action "scène"				
AS_Num		Numéro d'ordre de l'action "scène"	<i>num line</i>		GL_AS_AllData
AS_Desc		Description de l'action "scène"	(voir plus loin) items		GL_AS_AllRef
	Descr_Action	Structure unitaire de description d'une action			

TABLEAU 20

PS	Liste des phrases attachées à la scène				
	Pour chaque phrase gérée		fId PH_AllData	1 ligne	PS ReadListPh
	PH_Ident	Identifiant de la phrase – Data_ID = 1300 + PH_Ident		item 1	GL_PS_AllData
	PH_Origine	<pilote> Origine de la donnée: <pilote> (READ ONLY seulement)		item 2	GL_PS_AllText
	PH_Type	<txt> Type de la phrase ⇒ jrs <txt>		item 3	
	PH_Name	txt Nom de la phrase		item 4	
	Pour chaque ligne de texte	(caractère réservé pour "saut de ligne" (§))	fId PH_AllText	1 ligne	
	PH_Line	Lignes de texte			

TABIEAU 21

Structures unitaires									
Descr_Objet		Description d'un objet (scène ou appli)			OC_AllData				
Obj_Ident		Identifiant de l'objet			item 1				
Obj_Level		Niveau (pour affichage : 1 derrière, last devant)			item 2				
Obj_Displ_Type	<Obj_Displ_List>	Type d'affichage de l'objet (Tirs <sta> si niveau Appli)			item 3				
Obj_Displ_Val		Attribut selon le type de Display			item 4			GetOneObj(SC_ID,Obj_Ident)	
Obj_Type	<Obj_Type_List>	Type de l'objet (voir tableau)			item 5			(0 => "AP")	
Obj_Name		Nom de l'objet (sans virgules)			item 6			GL_Obj_OneData	
Obj_Data		Attributs de l'objet (selon son type : 8 max)			items 7 à 14				

TABIEAU 22

Obj_Displ_list	Valeurs possibles pour l'attribut DISP d'un objet		Attributs		
	<sta>	Affichage direct à l'OpenScene, sans conditions	-		
	<dyn>	Affichage via les actions de type <obj> (bool : OpenScene)	Bool		
	<opt>	Affichage lorsque la position xxx du curseur est désignée	OP_Num		

TABLEAU 23

Obj_Type_list	Type d'un objet						
< pict>	Image bitmap						
< gra>	Graphique généré sans média externe						
< text>	texte						

TABLEAU 24

Valeur de Obj_Data en fonction de Obj_Type								
	ref	posX	posY	LargeMax	HautMax	Coul	Font	Transp
<pic>	ME_Ident	num pix	num pix	-	-	-	-	Bool
<gra>	-	num pix	num pix	num pix	num pix	Pal_Num	-	-
<text>	Data_ID	num pix	num pix	num pix	num pix	Pal_Num	FO_Ident	-

TABLEAU 25

Descr_EvAction	Description d'une ligne Event/Action (scène ou appli)		AA_AllData				
Act_Event	<EV_Ident>	Identifiant de l'événement déclenchant	item 1				
Act_EvPar	Ev param	Paramètre de l'événement si besoin (ex. KeyNav \Rightarrow Up)	item 2				
Act_Pass		Vraie si événement continue (valeur par défaut ?)	item 3				
Act_Ident	<AC_Ident>	Identifiant de l'action à déclencher (d'un type)	item 4				
Act_Par1		Param 1 de description de l'action (selon le type)	item 5			GetOneEva(SC_ID,j)	
Act_Par2		Param 2 de description de l'action (selon le type)	item 6			(0 \Rightarrow "AP")	
<i>Pour chaque condition (ET)</i>			k			GL_Eva_OneData	
Act_CondRef		Référence à une condition	item k + 6				
<i>Structure interne : une ligne par CondRef</i>			AA_AllRef			GetOneCond(k)	
Act_CondRef		Identifiant de la condition	item 1			GL_Eva_OneRef	
Act_DataID	<Data_ID>	DO_Ident ou VA_Ident ou PH_Ident selon 1er chiffre	item 2			(via GL_Eva_OneData)	
Act_Value		Valeur qui doit avoir la VarData pour exécuter l'action	item 3				
Règle de gestion :			Act_Pass=false si Action = "GoScene", libre sinon				

TABLEAU 26

Nom touches Correspondant	Événement
TV	KeyTV
Quitter	KeyQuit
Veille	KeyVeille
Prefs	KeyPrefs
Liste	KeyListe
Date	KeyDate
P+	KeyP+
P-	KeyP-
Volume+	KeyVol+
Volume-	KeyVol-
Mute	KeyMute/unMute
Guide	KeyGuide
Info	KeyInfo
Ret	KeyBack
Haut	KeyUp
Bas	KeyDown
Droit	KeyRight
Gauche	KeyLeft
OK	KeyOK
Menu	KeyMenu
Message	Key@
Radio	KeyRadio
0 à 9	KeyNumb de 0 à 9
A	KeyA
B	
C	
D	
--	
Z	KeyZ

TABLEAU 27

1, <décodeur>, <num>, TheKeyNum0 exemple	--Ne pas toucher cette ligne données prédéfinies et
2, <appli>, <num>, TheCurOpt, 0 exemple	--Ne pas toucher cette ligne données prédéfinies et
3, <pilote>, <txt>, TheCurTime, 00h00 exemple	--Ne pas toucher cette ligne données prédéfinies et
4, <ext>, <Num>, TheData0, 0	
5, <ext>, <Num>, TheData1, 1	
6, <ext>, <Num>, TheData2, 2	
7, <ext>, <Num>, TheData4, 4	
8, <ext>, <Num>, TheData5, 5	
9, <ext>, <Num>, TheData6, 6	
10, <ext>, <Num>, TheData7, 7	
11, <ext>, <Num>, TheData8, 8	
12, <ext>, <Num>, TheData9, 9	

TABLEAU 28

Scènes niveau 0	Niveau 1	Niveau 2	Niveau 3	Niveau 4	Nombre de modules à acquérir	Total mémoire nécessaire
Scène 1 Média1:1 Média2:1 Média3:1	Scène 2 Média1:1 Média2:1 Média3:1 Scène 3 Média1:1 Média2:1 Média3:1 Scène 4 Média1:1 Média2:1 Média3:1	Scène 5 Média1:0 Média2:0 Média3:0	Scène 6 Média1:0 Média2:0 Média3:0	Scène 7 Média1:0 Média2:0 Média3:0	12	12x30
Scène 2 Média1:2 Média2:2 Média3:2	Scène 1 Média1:1 Média2:1 Média3:1	Scène 3 Média1:0 Média2:0 Média3:0 Scène 4 Média1:0 Média2:0 Média3:0	Scène 5 Média1:0 Média2:0 Média3:0	Scène 6 Média1:0 Média2:0 Média3:0	0	6x30
Scène 3 Média1:2 Média2:2 Média3:2	Scène 1 Média1:1 Média2:1 Média3:1	Scène 2 Média1:0 Média2:0 Média3:0 Scène 4 Média1:0 Média2:0 Média3:0	Scène 5 Média1:0 Média2:0 Média3:0	Scène 5 Média1:0 Média2:0 Média3:0 Scène 7 Média1:0 Média2:0 Média3:0	0	
Scène 5 Média1:2 Média2:2 Média3:2	Scène 6 Média1:1 Média2:1 Média3:1 Scène 7 Média1:1 Média2:1 Média3:1	Scène 1 Média1:0 Média2:0 Média3:0	Scène 4 Média1:0 Média2:0 Média3:0		6	9x30

Scène 6 Média1:2 Média2:2 Média3:2	Scène 5 Média1:1 Média2:1 Média3:1 Scène 7 Média1:1 Média2:1 Média3:1	Scène 1 Média1:0 Média2:0 Média3:0	Scène 3 Média1:0 Média2:0 Média3:0 Scène 2 Média1:0 Média2:0 Média3:0		0	
			Scène 4 Média1:0 Média2:0 Média3:0			
Scène 7 Média1:2 Média2:2 Média3:2	Scène 1 Média1:1 Média2:1 Média3:1	Scène 3 Média1:0 Média2:0 Média3:0 Scène 2 Média1:0 Média2:0 Média3:0 Scène 4 Média1:0 Média2:0 Média3:0	Scène 5 Média1:0 Média2:0 Média3:0			

REVENDEICATIONS

1. Système de développement d'applications interactives distribuées sur un canal de transmission numérique, ces applications interactives installées en un point
5 du réseau de transmission numérique consistant en une succession de scènes thématiques incorporant au moins des images en mode point, des images vidéo, des séquences sonores et des informations sous forme de fichier texte représentés au niveau d'au moins un moniteur d'affichage, le
10 caractère interactif de ces applications étant produit par l'intermédiaire d'une fonctionnalisation spécifique de l'interface de commande d'un module de commande du moniteur d'affichage muni d'un système d'exploitation, l'ensemble moniteur d'affichage, module de commande et système
15 d'exploitation constituant une plate-forme d'exécution, caractérisé en ce qu'il comporte au moins :

- un éditeur électronique permettant la simulation, d'une part, de l'écran d'affichage dudit moniteur d'affichage, et, d'autre part, de l'interface de commande
20 d'un module de commande associé à ce moniteur d'affichage ;

- des moyens générateurs d'objets virtuels, spécifiques à l'application considérée, chaque objet virtuel constitué par des composants choisis parmi au moins des
25 images en mode point, des images vidéo, des séquences sonores et des informations sous forme de fichiers textes étant susceptible d'être édité, affiché et représenté par ledit éditeur électronique pour former au moins une scène constitutive de ladite application ;

30 - des moyens de commande de mise en corrélation d'une pluralité de couples événement/action, un événement

étant défini comme la rupture d'un équilibre d'état dudit système d'exploitation et une action étant définie comme une initialisation respectivement une modification d'au moins un des paramètres d'affichage et de représentation d'une ou plusieurs scènes constitutives de ladite application ;

des moyens de mémorisation de l'ensemble des couples événement/action, mis en corrélation et constituant ladite application interactive.

10 2. Système selon la revendication 1, caractérisé en ce que lesdits moyens de commande de mise en corrélation d'une pluralité de couples événement/action, comportent des moyens générateurs d'une série d'instructions présentant une structure de contrôle de type liste de re-

15 quêtes pour "EVENEMENT" IF condition sur toute variable booléenne $B_i \in [B_0, B_1, \dots, B_j, B_N]$ ensemble de variables booléennes vraie THEN "ACTION" où "EVENEMENT" désigne une variable représentative d'un événement et "ACTION" désigne une variable représentative d'une action conditionnée à

20 l'ensemble des variables booléennes, ladite variable "EVENEMENT" vraie constituant pour chaque couple événement/action une rupture d'équilibre d'état dudit système d'exploitation et ladite action constituant une initiali-

25 sation ou une modification d'au moins un des paramètres d'affichage et de représentation d'une ou plusieurs scènes constitutives de ladite application.

3. Système selon la revendication 2, caractérisé en ce que l'ensemble des couples événement/action est organisé selon une suite de lignes de macro-instructions

30 comprenant chacune un champ de position d'exécution, un

champ événement, un champ action, un champ liste de conditions et un champ variable booléenne d'arrêt.

4. Système selon l'une des revendications 1 à 3, caractérisé en ce que ledit moniteur d'affichage, ledit module de commande et ledit système d'exploitation étant ceux d'un récepteur de télévision, ledit éditeur électronique permet la simulation de l'écran d'affichage de ce récepteur de télévision et de l'interface de commande d'un module de télécommande universel associé à ce récepteur de télévision, à une touche spécifique dudit module de télécommande universel simulé étant associé un événement, provoquant la rupture d'équilibre d'état dudit système d'exploitation.

5. Système selon l'une des revendications 1 à 3, caractérisé en ce que ledit moniteur d'affichage, ledit module de commande et ledit système d'exploitation étant ceux d'une station de travail ou d'un micro-ordinateur, ledit éditeur électronique permet la simulation de l'écran d'affichage de ce micro-ordinateur et de l'interface de commande d'un appareil périphérique d'entrée de ce micro-ordinateur, tel que le clavier, à une touche spécifique dudit appareil périphérique d'entrée étant associé un événement provoquant la rupture d'équilibre d'état dudit système d'exploitation.

6. Système selon l'une des revendications 1 à 5, caractérisé en ce que ledit éditeur électronique, lesdits moyens générateurs d'objets virtuels spécifiques à l'application et lesdits moyens de commande de mise en corrélation d'une pluralité de couples événement/action comprennent un module logiciel pilote permettant, d'une part, à partir d'une fenêtre d'affichage d'une représenta-

tion dudit module de commande et d'appareils périphériques, de recevoir une variable "EVENEMENT" correspondante, et, d'autre part, à partir d'une fenêtre d'affichage de scènes constitutives de ladite application, de provoquer
5 soit l'initialisation, soit la modification de ce celles-ci.

7. Système selon la revendication 6, caractérisé en ce que celui-ci comporte en outre un outil de création de macro-instructions, ledit outil comportant au moins une
10 page écran de mise en corrélation de couples événement/action comprenant au moins une fenêtre d'affichage d'un ensemble d'événements, une fenêtre d'affichage d'une action déclenchée associée à l'un des événements de cet ensemble d'événements et au moins une fenêtre de paramé-
15 trage de l'action déclenchée, des moyens de pointage et de sélection, ladite mise en corrélation de couples événement/action étant effectuée par sélection d'un événement parmi l'ensemble d'événements et de l'action par l'intermédiaire du moyen de pointage.

20 8. Système selon la revendication 6 ou 7, caractérisé en ce qu'il comporte en outre un outil de création d'objets virtuels, ledit outil de création d'objets virtuels comportant au moins une page écran de création d'objets comprenant au moins une fenêtre d'affichage d'une
25 liste d'objets accessibles, cette liste d'objets étant affichée selon un champ de niveau et un champ de désignation d'objet, une fenêtre de choix de mode d'affichage/représentation de cet objet, un champ de type d'objet permettant de discriminer la nature de cet objet, un
30 champ relatif à un ensemble de paramètres fonctions du média d'affichage/représentation de cet objet.

9. Système selon l'une des revendications 6 à 8, caractérisé en ce qu'il comporte en outre un outil d'édition des médias, ledit outil d'édition des médias comportant au moins une fenêtre page écran "Média" comprenant au moins une fenêtre d'affichage d'une liste des plateformes d'exécution, une fenêtre d'affichage d'une liste de médias spécifiques susceptibles d'intervenir dans la composition d'une scène, chaque média étant défini par un nom et un attribut, l'attribut alloué à un média donné permettant de discriminer le caractère interne ou externe du média vis-à-vis de l'application considérée.

10. Système selon l'une des revendications 6 à 9, caractérisé en ce qu'il comporte en outre un outil d'édition de phrases sous forme de fichier texte, chaque phrase comportant un numéro d'identification permettant d'affecter cette phrase soit au niveau de l'application, soit au niveau d'une ou plusieurs scènes constitutives de cette application.

11. Système selon l'une des revendications 6 à 10, caractérisé en ce qu'il comporte en outre un outil de création de scène comportant au moins une fenêtre d'affichage d'une liste de scènes types repérées par une désignation spécifique, une fenêtre de définition de paramètres de la scène choisie, paramètres tels que numéro d'identification de scène, palette de couleurs, choix du fond à l'initialisation de la scène, position de départ du curseur, et une fenêtre de choix d'options, ledit outil permettant de créer une nouvelle scène, de dupliquer ou de supprimer une scène existante.

12. Système selon l'une des revendications 6 à 11, caractérisé en ce qu'il comporte en outre un outil de dé-

finition d'options dans une scène, ledit outil de définition d'options permettant à partir d'une liste d'options définie pour la scène considérée,

- d'attribuer une option déterminée parmi ladite
 5 liste d'options, chaque option correspondant à un attribut
 d'affichage d'un objet virtuel ; et

- de naviguer et choisir une option auxiliaire autour de l'option attribuée à partir d'un ensemble de boutons de commande du module de commande associé au moniteur
 10 d'affichage.

13. Système selon l'une des revendications 6 à 12, caractérisé en ce qu'il comporte en outre un outil d'édition des attributs d'une application déterminée, lesdits attributs comportant au moins un champ commentaires descriptifs d'application, un champ auteur de l'application,
 15 un champ de nom de l'application, un champ de numéro de version de l'application, un champ de palette de couleurs de l'application.

14. Système selon l'une des revendications 6 à 13, caractérisé en ce que, pour chaque plate-forme d'exécution d'application, celui-ci comporte un ensemble de tables de données communes au couple plate-forme d'exécution/application, ledit ensemble de données communes comportant au moins :

25 - une table des données d'application relatives à ladite plate-forme ;

- une table des données caractéristiques de chaque plate-forme ;

- une table des événements relatifs à ladite
 30 plate-forme ;

- une table des actions d'application relatives à ladite plate-forme ;

- une table des fontes disponibles pour ladite plate-forme.

5 15. Système selon la revendication 14, caractérisé en ce que ladite table des données d'application relatives à ladite plate-forme est définie pour l'application considérée par l'intermédiaire d'un fichier texte.

10 16. Système selon l'une des revendications 1 à 15, caractérisé en ce que chaque application est définie par un ensemble de listes comportant au moins une liste des attributs d'application, une liste des objets virtuels communs, une liste des couples événement/action, une liste des fontes, une liste des médias, une liste des variables,
15 une liste des phrases, une liste des palettes de couleurs, une liste des couleurs et une liste des scènes.

17. Système selon la revendication 16, caractérisé en ce que chaque élément de la liste des scènes, relatif à une scène considérée, constitutive de cette application,
20 comporte au moins une liste d'attributs de scène, une liste des objets virtuels constitutifs de la scène, une liste d'options, une liste des phrases de la scène.

18. Système selon l'une des revendications 1 à 5, caractérisé en ce que l'ensemble des structures de données
25 et de macro-instructions constituant une application interactive native, celui-ci comporte en outre un module de traduction de ladite application interactive native en une application interactive dédiée à un terminal d'accès de type déterminé.

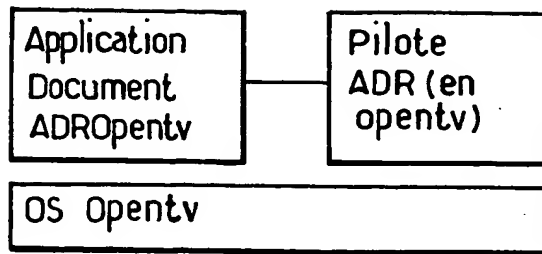
30 19. Système selon l'une des revendications 1 à 18, caractérisé en ce que, dans le but de développer des ap-

plications interactives à diffusion cyclique accessibles sur un canal TV à partir d'un terminal d'accès muni d'un décodeur TV, ledit système comporte au moins :

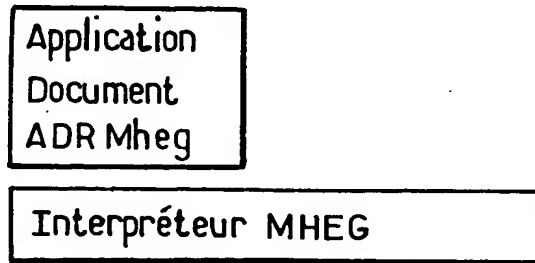
- un moyen gestionnaire d'une pluralité de scènes successives constitutives de cette application selon un niveau de profondeur de chaque scène vis-à-vis de la scène courante, ledit module gestionnaire permettant d'analyser et de discriminer sur un nombre déterminé de scènes successives voisines de la scène courante les objets virtuels communs et les médias communs associés à ces objets de ces scènes successives des objets virtuels distincts et des médias distincts associés à ces objets de ces scènes successives ;

- des moyens d'actualisation desdits objets virtuels et médias communs et des objets virtuels et médias distincts pour au moins une scène successive constitutive de cette pluralité de scènes successives par rapport à la scène courante, ce qui permet d'optimiser le volume de données de réactualisation et le temps d'acquisition de ces données d'actualisation en fonction de la taille mémoire allouée à chaque type de média et du nombre de niveaux de profondeur d'analyse de chaque scène successive.

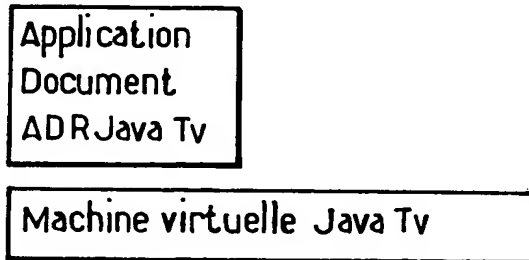
1.) OPentv



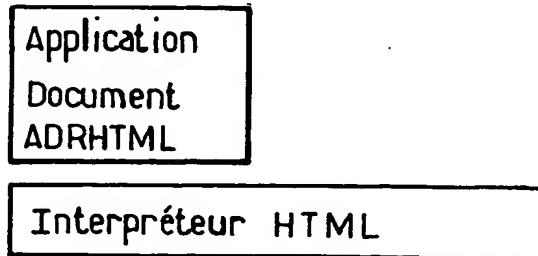
2.) Mheg



3.) Java Tv



4.) Application HTML



5.) Application Document pour logiciel spécifique pour WWW (exemple Roaster, Neuron) Plateforme internet

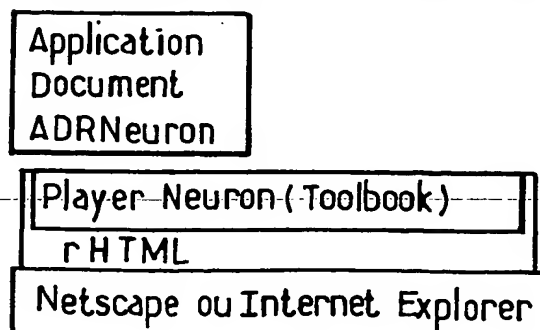


FIG.1a.

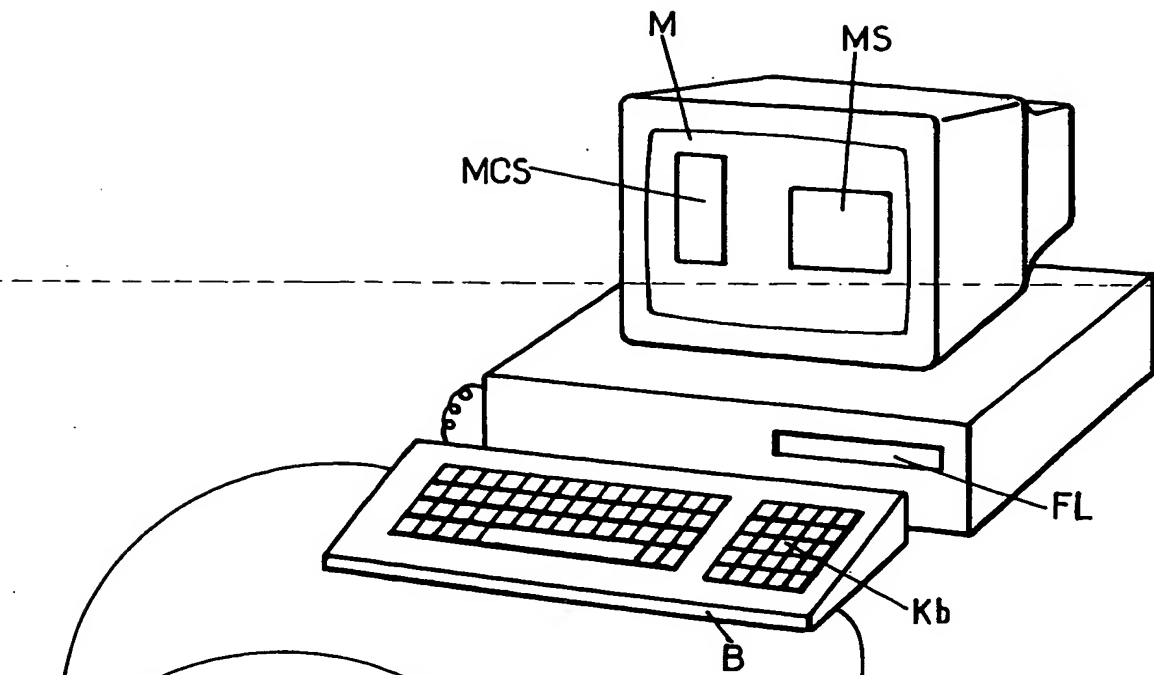


FIG. 1b.

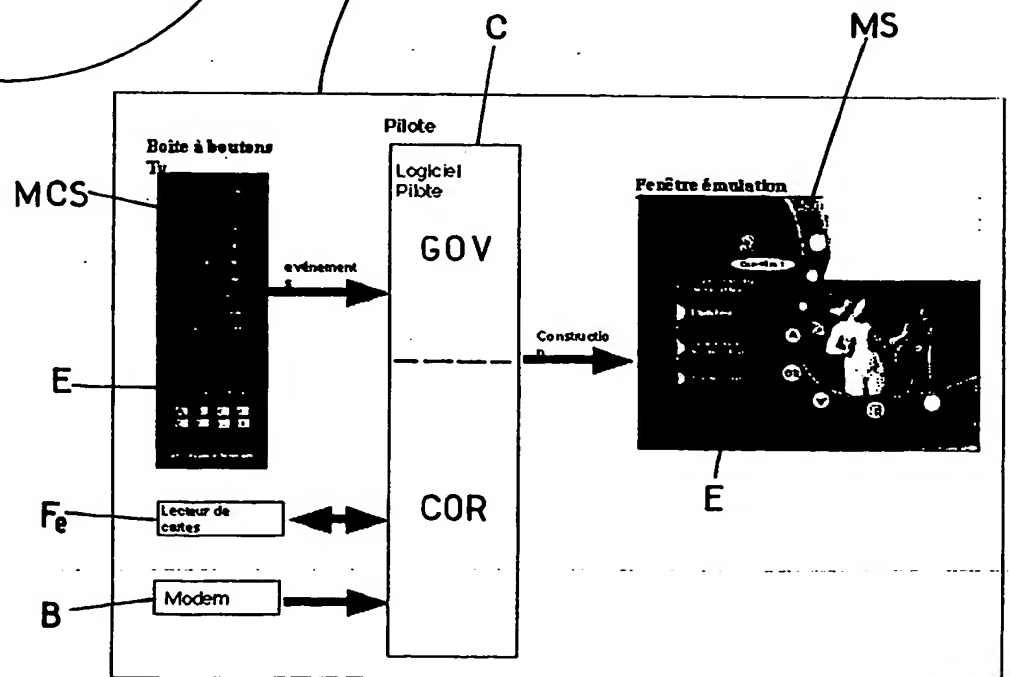
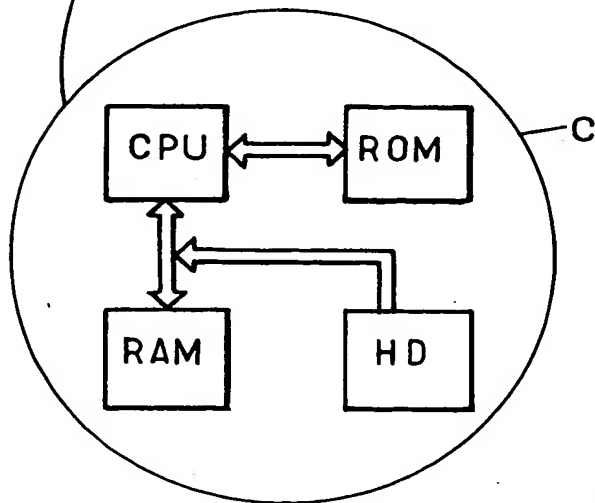
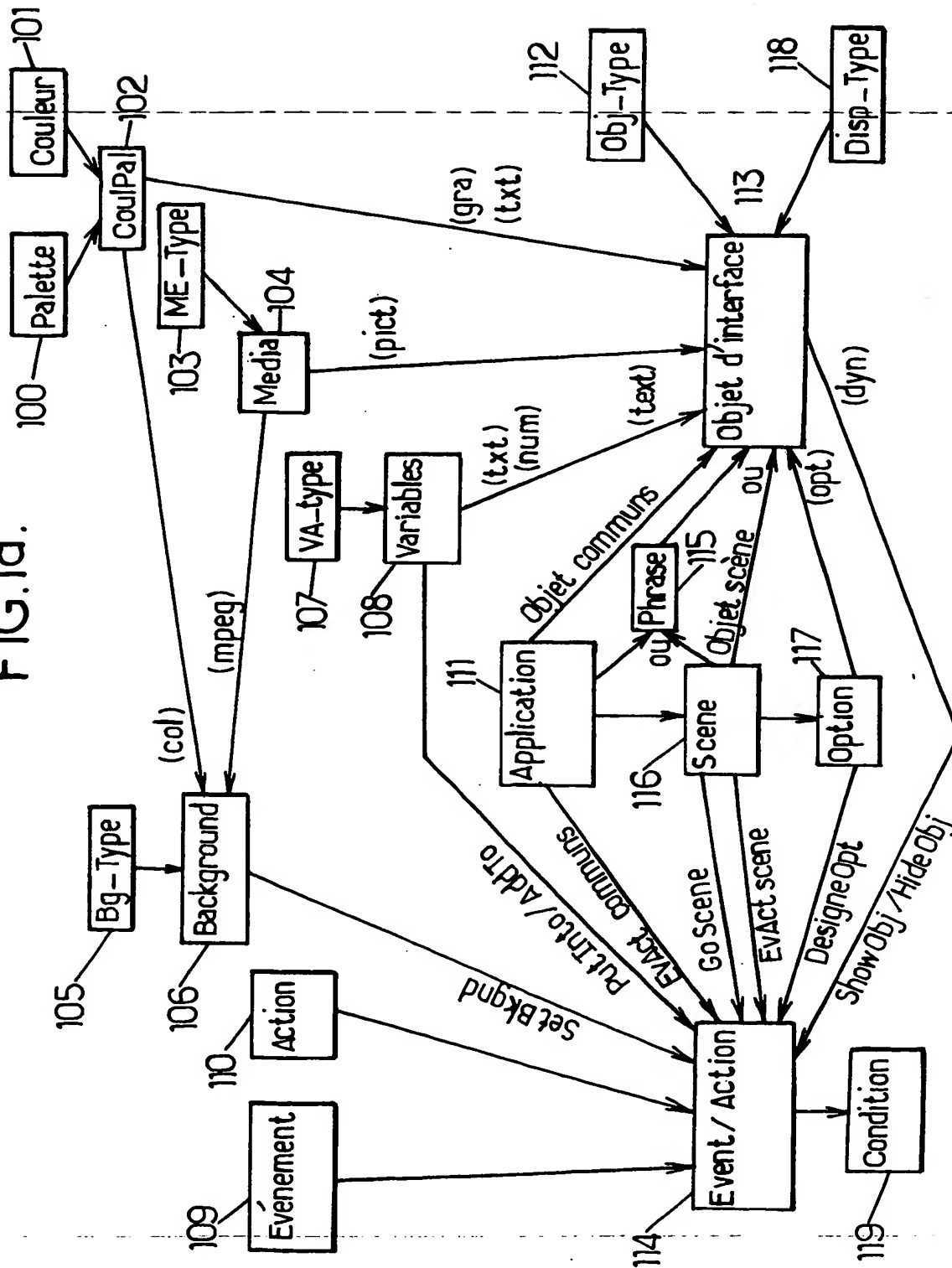


FIG. 1d.



Structure d'une application

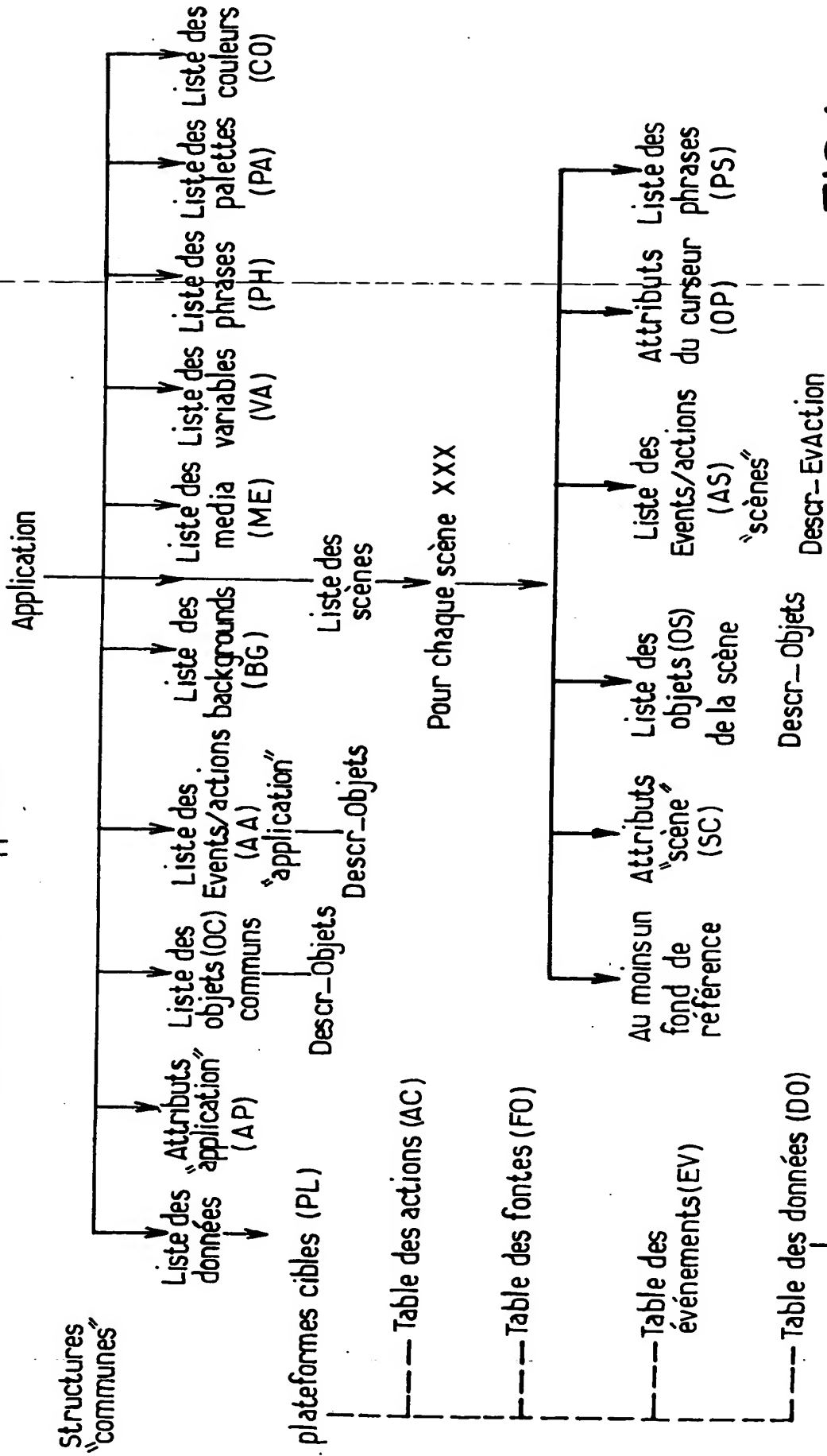


FIG.1c.

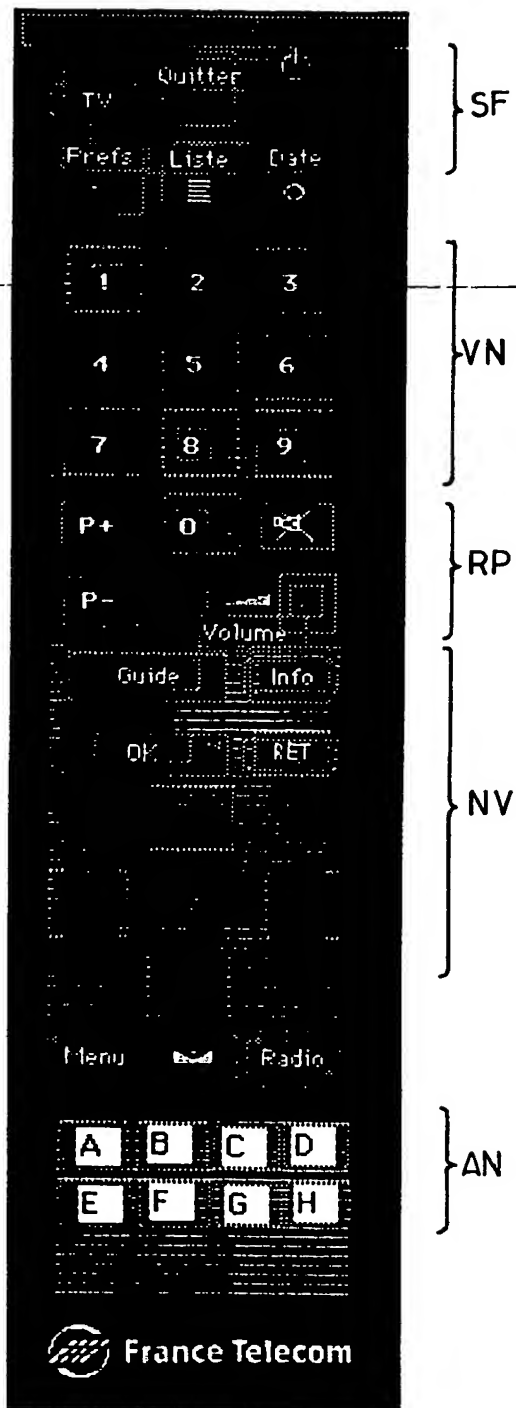


FIG.2.
MODULE COMMANDE T V
SIMULÉ

PalEvActions

Liste des Evénements/Actions Q1

Liste des Ev/Actions:
 KeyNum 1
 SceneOpen
 TimerExpired

Action déclenchée: GoScene

Param 1: 21

Param 2:

Conditions de déclenchement:

Niveau: 1

Ajouter Supprimer

Donnée/Variable:
 Valeur:

☐ Continuer dans la liste

Ligne de script 1

2,1,false,6,126,-1
7,-1,true,13,0,1,-1
8,-1,true,1,5,-1

FIG.3a1.

PalEvActions

Liste des Evénements/Actions Q1

Liste des Ev/Actions:
 KeyNum 1
 SceneOpen
 TimerExpired

Action déclenchée: SetTimer

Param 1: 0.1 sec

Param 2:

Conditions de déclenchement:

Niveau: 2

Ajouter Supprimer

Donnée/Variable:
 Valeur:

☒ Continuer dans la liste

Click to display the data of this Event/Action

Ligne de script 2

2,1,false,6,126,-1
7,-1,true,13,0,1,-1
8,-1,true,1,5,-1

FIG.3a2.

PalEvActions

Liste des Evénements/Actions Q1

Liste des Ev/Actions:
 KeyNum 1
 SceneOpen
 TimerExpired

Action déclenchée: SetBg

Param 1: fond1

Param 2:

Conditions de déclenchement:

Niveau: 3

Ajouter Supprimer

Donnée/Variable:
 Valeur:

☒ Continuer dans la liste

Click to display the data of this Event/Action

Ligne de script 3

2,1,false,6,126,-1
7,-1,true,13,0,1,-1
8,-1,true,1,5,-1

FIG.3a3.

Définition de l'objet 1:

FIG.3b1.

CC-Audit	
1,1	<sta>-1,<text>q1,1301,213,147,300,2,2,1,1
2,2	<sta>-1,<text>titre,1302,110,187,250,2,3,1,1
3,3	<opt>1,<pict>curs1,7,57,239,30,30,1,1,false
4,4	<sta>-1,<text>opt1,1303,110,257,100,1,3,1,1
5,5	<opt>2,<pict>curs2,7,319,30,30,1,1,false
6,6	<sta>-1,<text>opt2,1304,110,326,150,2,3,1,1
7,7	<opt>3,<pict>curs3,8,57,389,30,30,1,1,false
8,8	<sta>-1,<text>opt3,1305,110,406,250,1,3,1,1
9,9	<sta>-1,<pict>fhaut,13,315,305,30,30,1,1,false
10,10	<sta>-1,<pict>fok,10,309,394,30,30,1,1,false

Liste des objets d'interface Q1

Liste des objets:

- 1 q1
- 2 titre
- 3 curs1
- 4 opt1
- 5 curs2
- 6 opt2
- 7 curs3
- 8 opt3
- 9 fhaut
- 10 ok
- 11 fbas
- 12 loupe

Derrière

Niveau: 1

Devant

Type de l'objet: <text>

Mode d'affichage: <sta>

Toujours visible

Phrase so q1

Helvetica 18

Pos X: 213

Pos Y: 147

Hauteur: 2

Lignes

largeur: 300

Ajoute

Duplique

Supprime

Cliquer sur une scène dans la liste des scènes pour afficher ses objets

Définition de l'objet 3:

FIG.3b2.

CC-Audit	
1,1	<sta>-1,<text>q1,1301,213,147,300,2,2,1,1
2,2	<sta>-1,<text>titre,1302,110,187,250,2,3,1,1
3,3	<opt>1,<pict>curs1,7,57,239,30,30,1,1,false

Liste des objets d'interface Q1

Liste des objets:

- 1 q1
- 2 titre
- 3 curs1
- 4 opt1
- 5 curs2
- 6 opt2
- 7 curs3
- 8 opt3
- 9 fhaut
- 10 ok
- 11 fbas
- 12 loupe

Derrière

Niveau: 3

Devant

Type de l'objet: <pict>

Mode d'affichage: <opt>

opt 1

Media: curs1

Pos X: 57

Pos Y: 239

Hauteur

largeur

Ajoute

Duplique

Supprime

Click to display the data of this object (and flash it on the TVset screen)

ME AllData	ME Allname	BG AllData
1.<mpeg>,ap,true,ext,1,2	accueil	1.<video>,-1,1,Fond vidéo
2.<mpeg>,ap,0e,ext,3,4	explic	2.<mpeg>,1,-1,fondaccueil
3.<mpeg>,SC,false,ext,5,6	fond1	3.<mpeg>,2,-1,explic
4.<mpeg>,SC,false,ext,7,8	rep1	4.<mpeg>,46,-1,fondal
5.<mpeg>,SC,false,ext,9,10	zoom1	5.<mpeg>,3,-1,fond1
6.<mpeg>,ap,true,ext,11,12	resum	6.<mpeg>,47,-1,rep1
7.<pict>,ap,true,ext,13,14	curs1	

ME AllRef
1,1,pct,128,128,adrmpeg:accueil.pct
2,2,mpg,60,60,accueil.mpg
3,1,pct,128,128,adrmpeg:explic.pct
4,2,pix,30,10,explic.mpg

Media Appli

Scènes Couleurs Variables Media Bkgnds Palettes Application

Liste des média :

- 1 accueil
- 2 explic
- 3 fond1
- 4 rep1
- 5 zoom1
- 6 resum

Liste des plateformes :

- Macintosh
- Sagem câble de

Media ID: 1
 Type du média: <mpeg>
 Chargement du média: ap
 Quitte si AP et pas présent: true
 Editable en exploitation: ext

Référence ID: 1
 Format du média source: pct
 Taille physique du média: 128
 Place mémoire vive: 128

Référence pilote: adrmpeg:accueil.pct

Ajoute
 Utilisé...

FIG.3c.

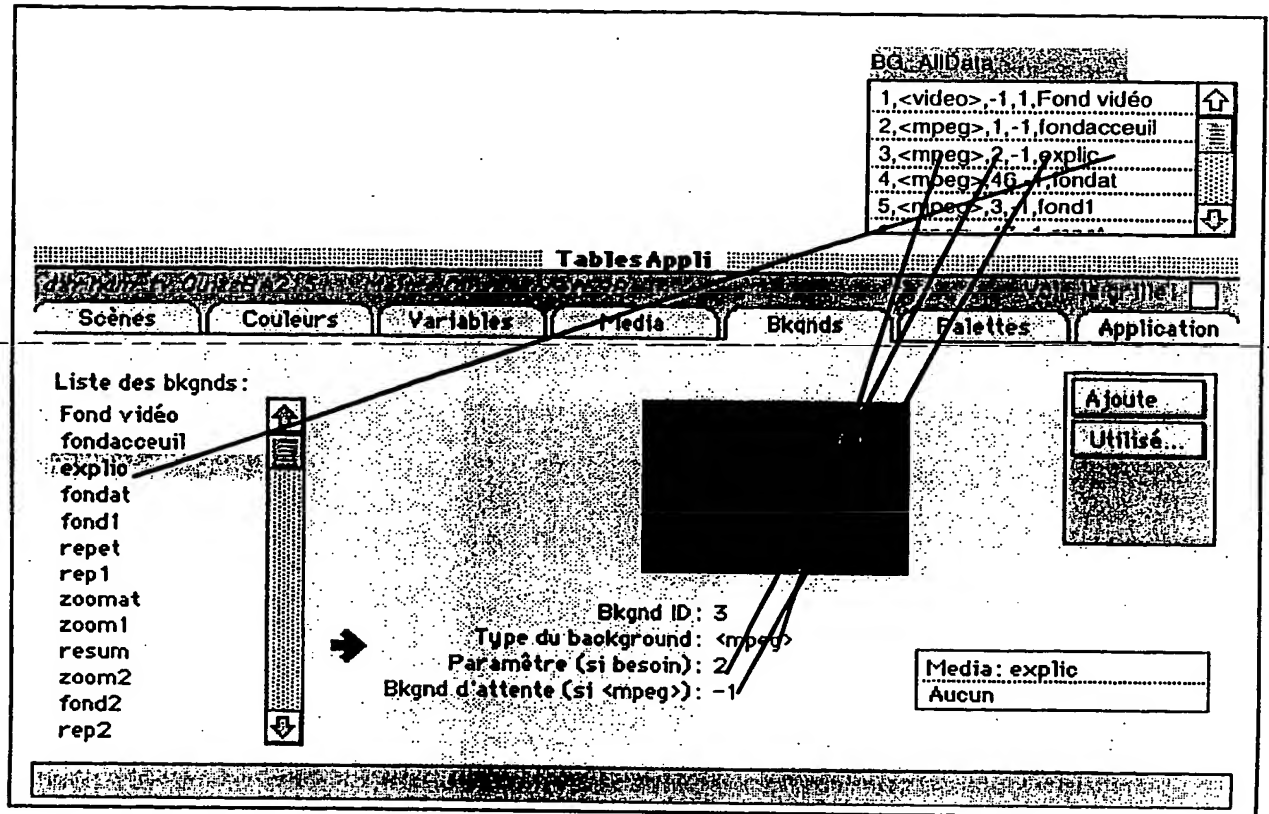
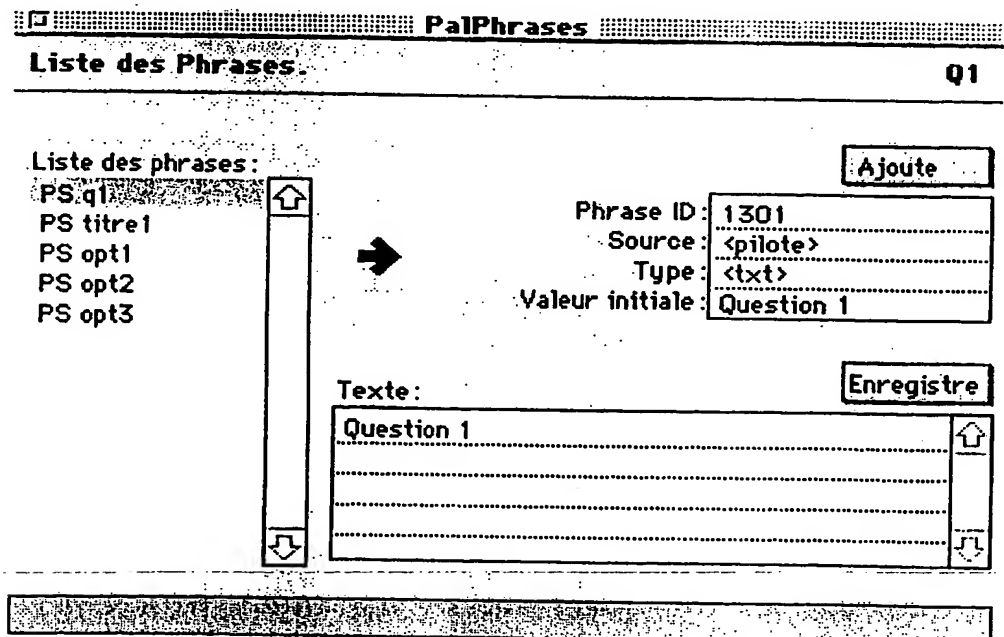


FIG.3d.

FIG.3e.



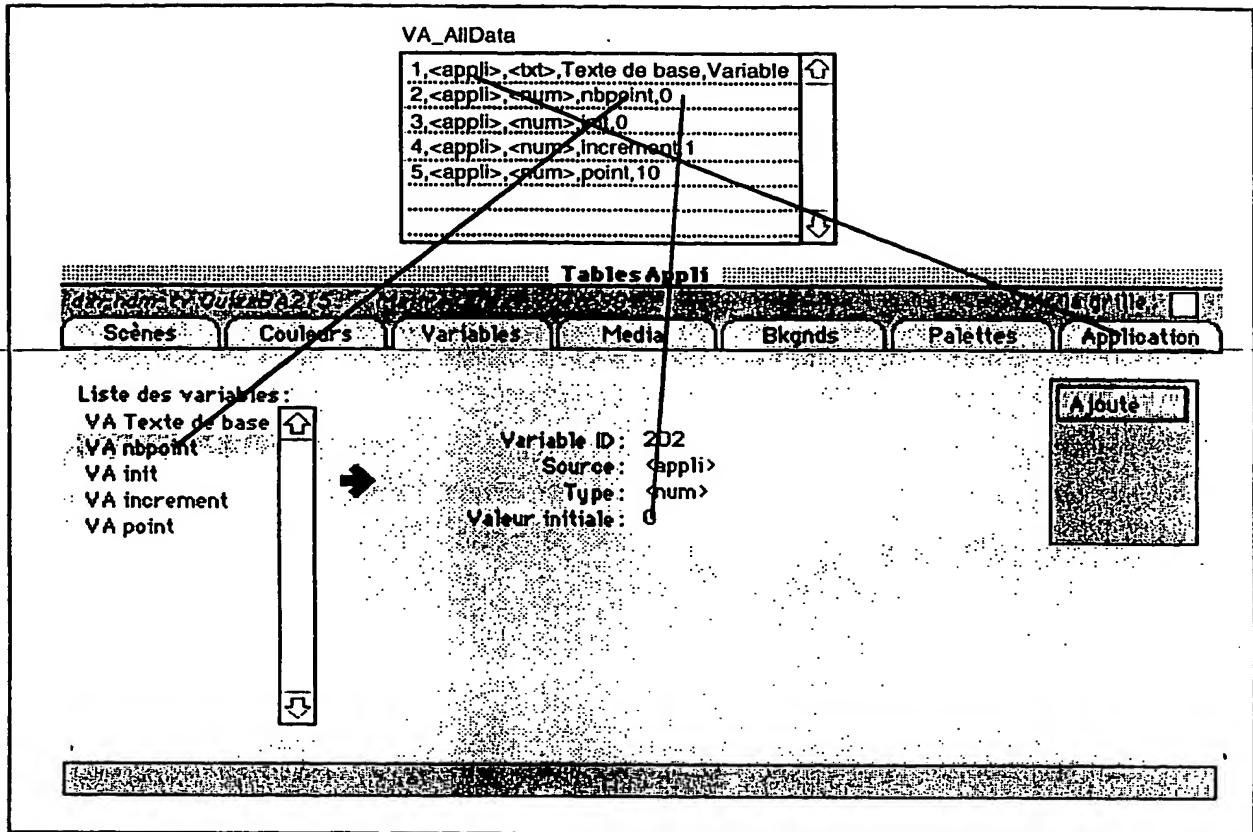


FIG. 3f.

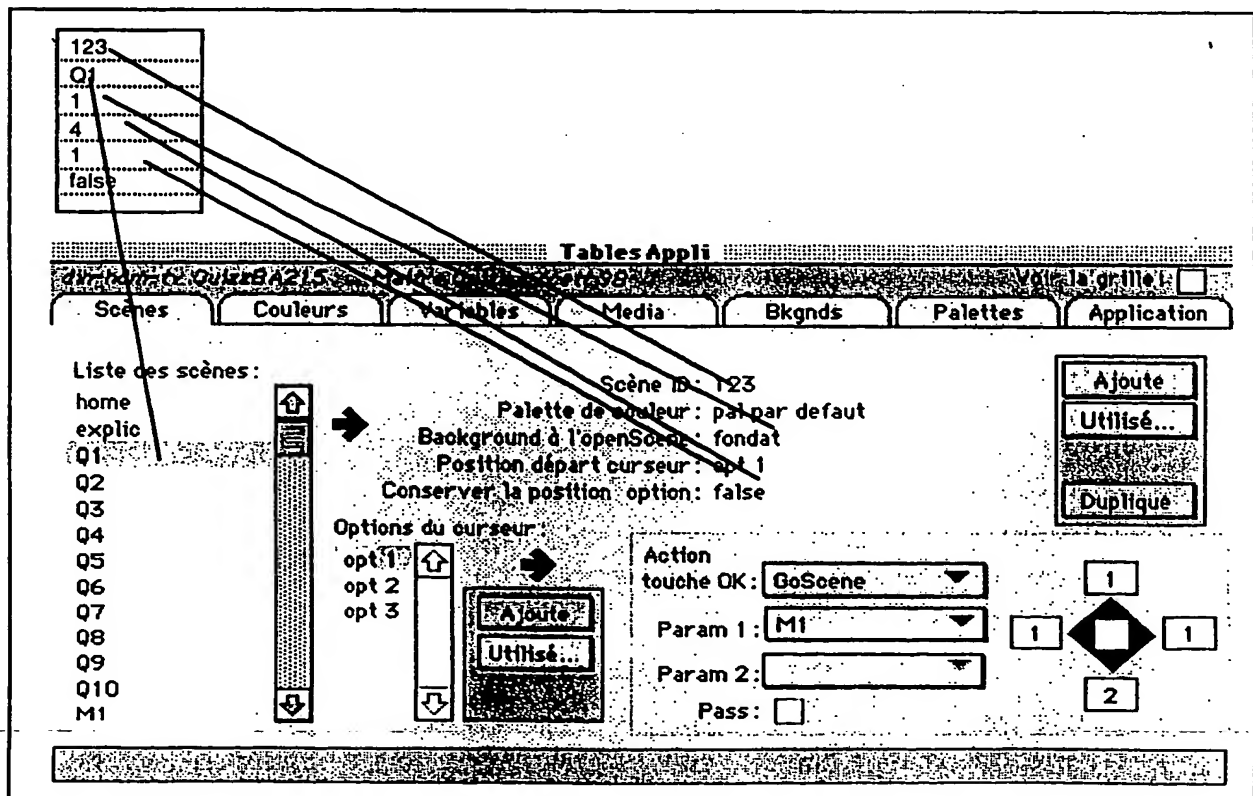


FIG. 3g.

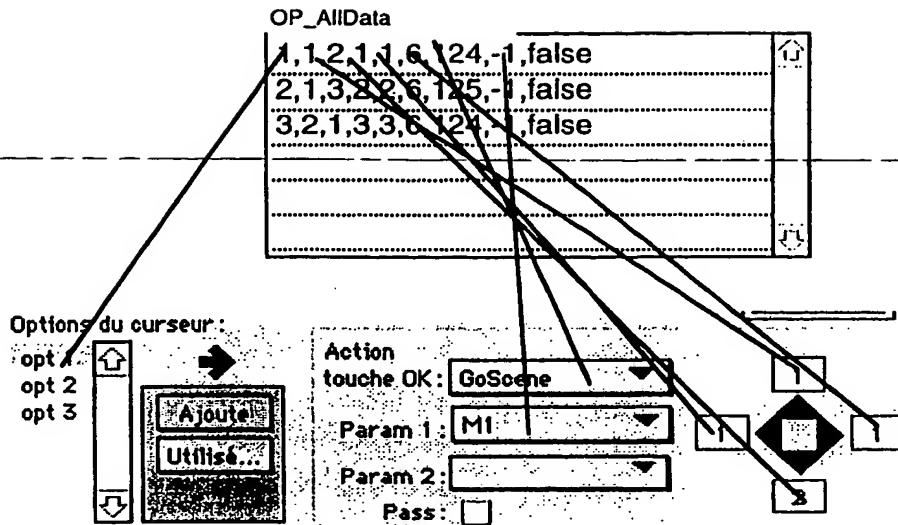


FIG.3h1.

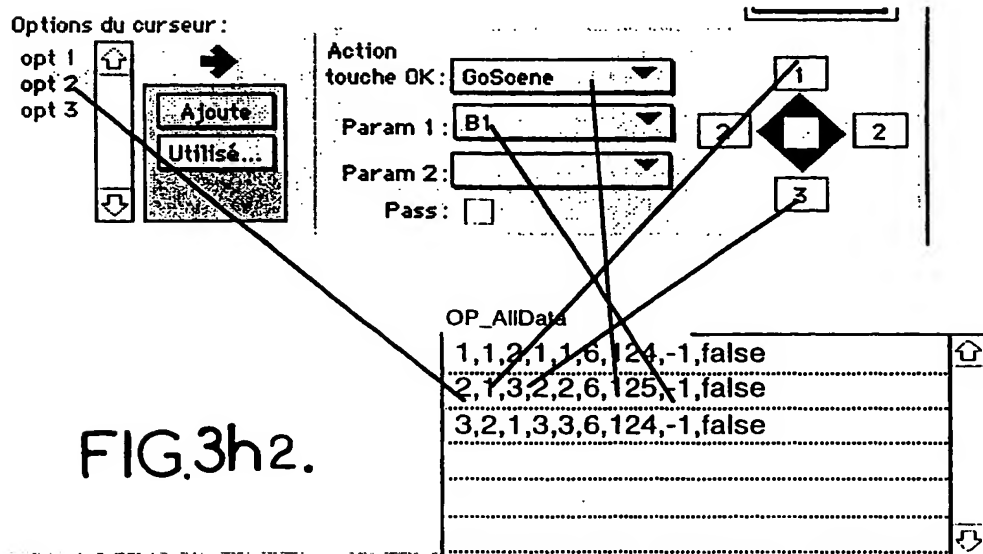


FIG.3h2.

The image shows a software configuration window titled "Tables Appli". It features several tabs: "Scènes", "Couleurs", "Variables", "Media", "Bgnds", "Palettes", and "Application". The "Application" tab is currently selected. Above the main window, there are two external boxes. The left box, titled "AP/OnData", contains a list of values: "test", "v1.0 alpha", "1", "-1", "31", and "true, true". The right box, titled "AP/OnData", contains "quizz beaux arts du 25/08/98" and "CNET-DIH-HDM-DEI/YH-JAC". Arrows point from these boxes to specific fields in the main window. The main window has a "Position écran" label pointing to a field. The "Application" tab contains the following fields:

- Nom de l'application: test
- Numéro de version: v1.0 alpha
- Palette par défaut: al par défaut
- Donnée pour la scène départ: aucune (Home)
- Couleur de transparence: coul_31
- Génération plateformes:
 - Macintosh => true
 - Sagem côle dev => true
- Description: quizz beaux arts du 25/08/98
- Auteur, copyright...: CNET-DIH-HDM-DEI/YH-JAC
- Enregistre

FIG.3i.

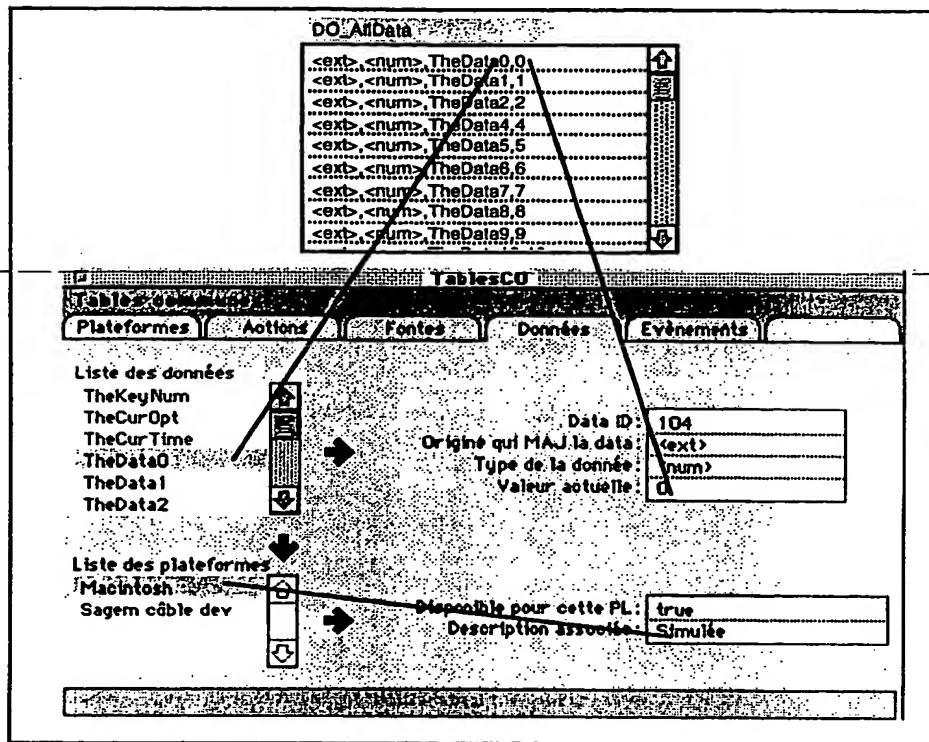


FIG. 3j1.

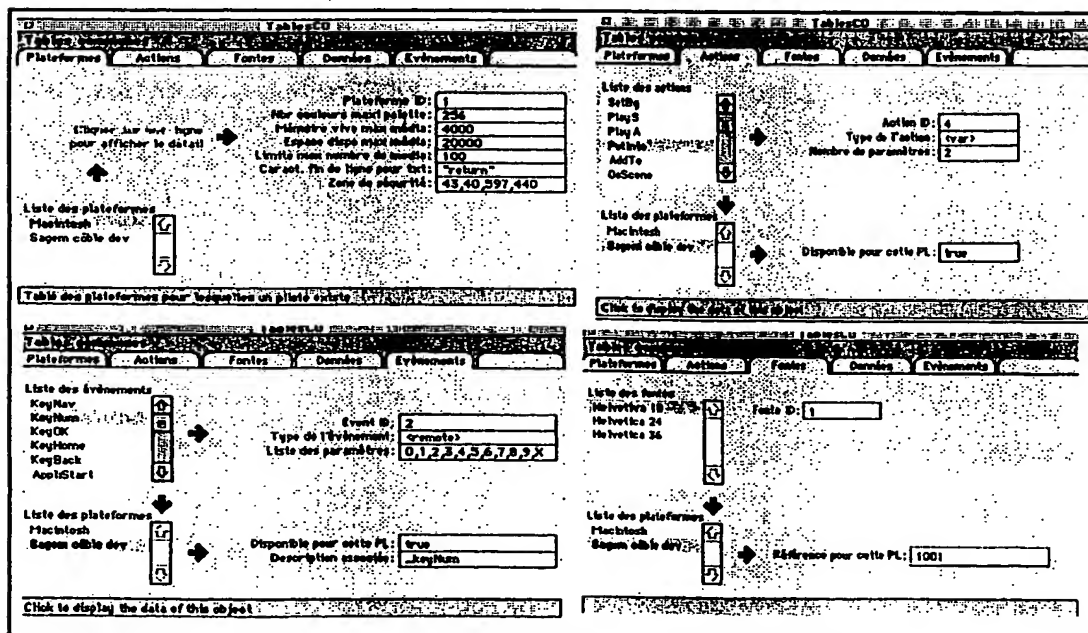


FIG. 3j2.

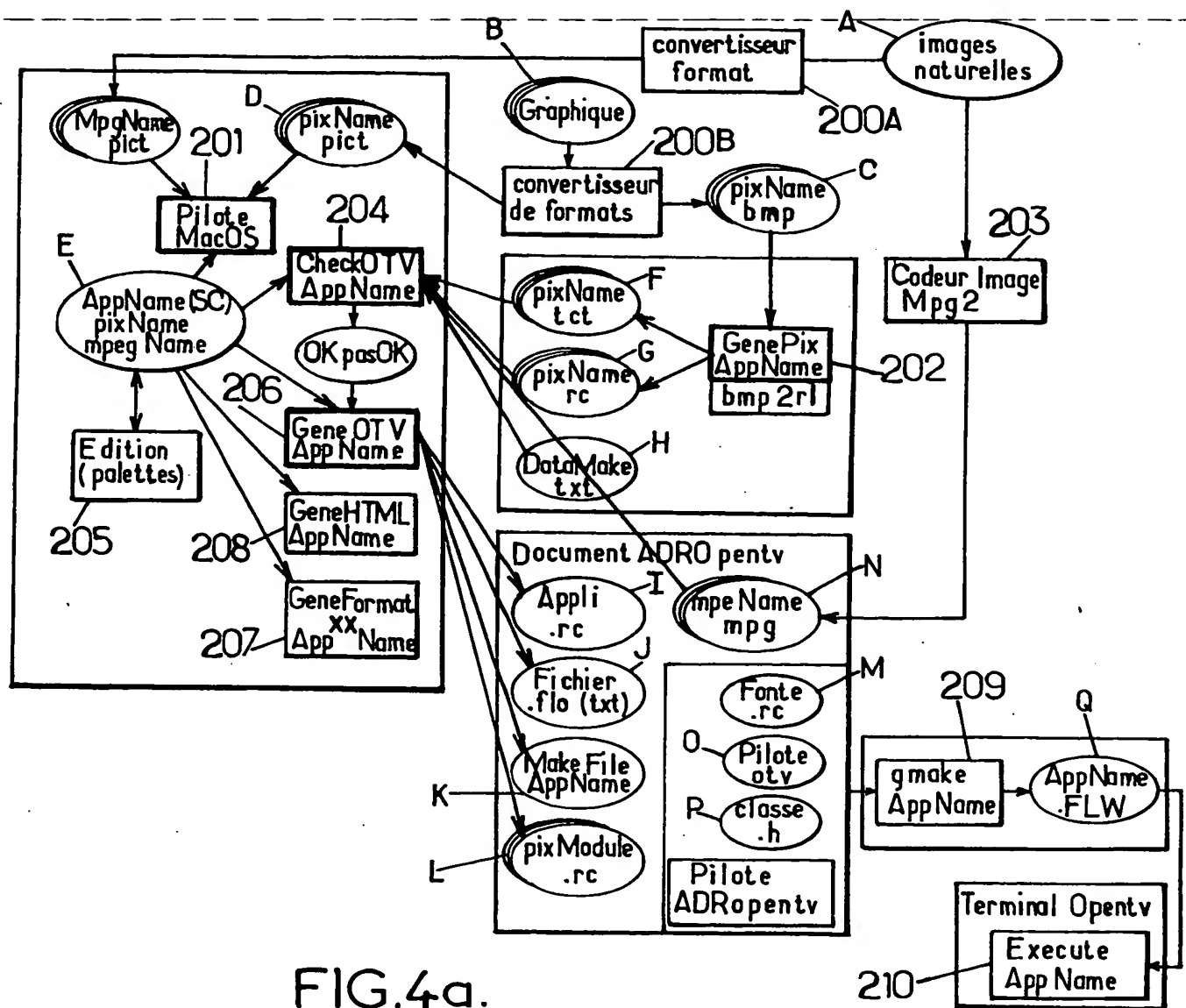


FIG.4a.

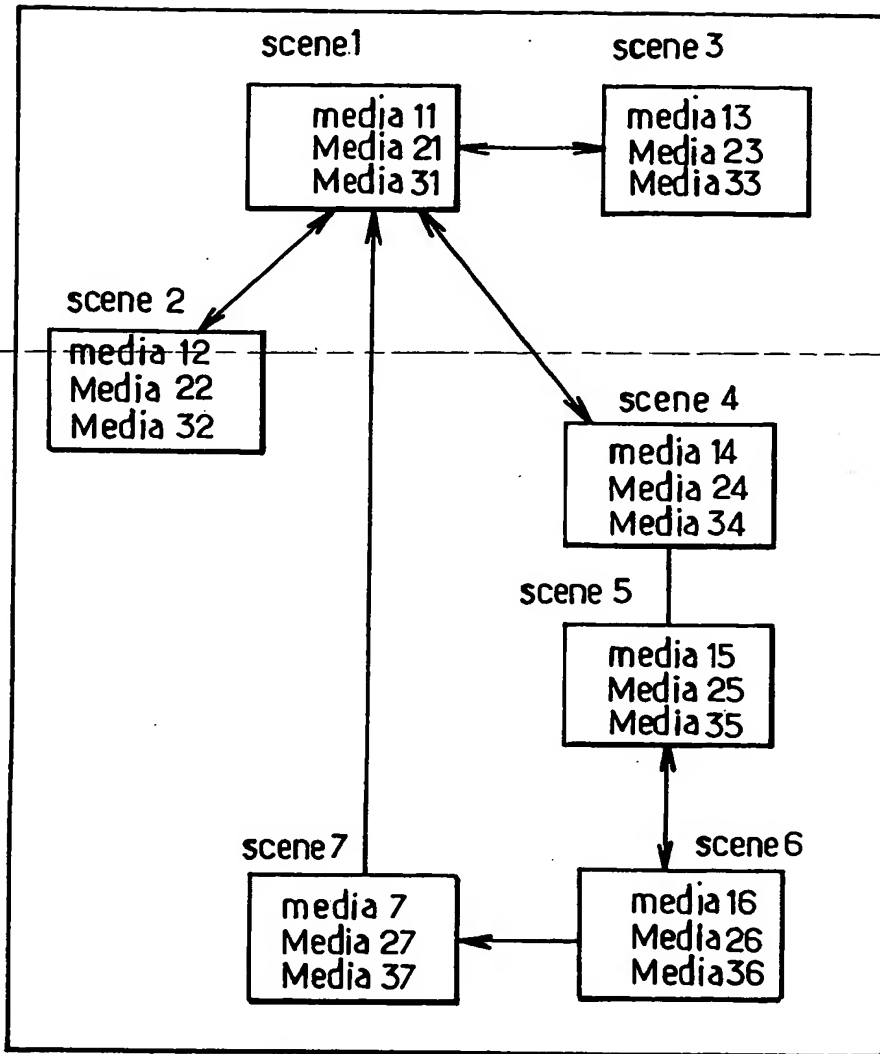


FIG.5b.

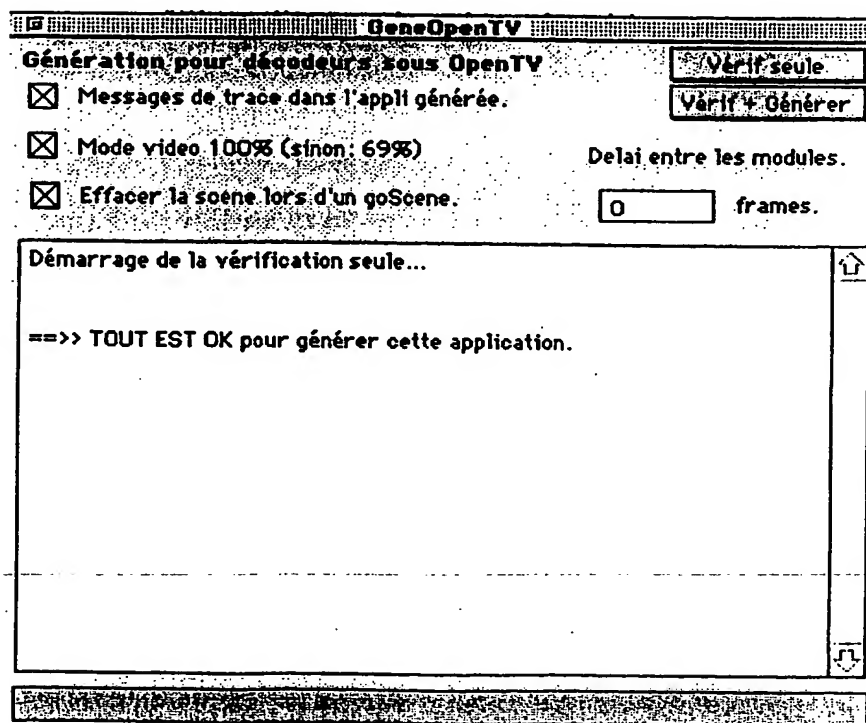
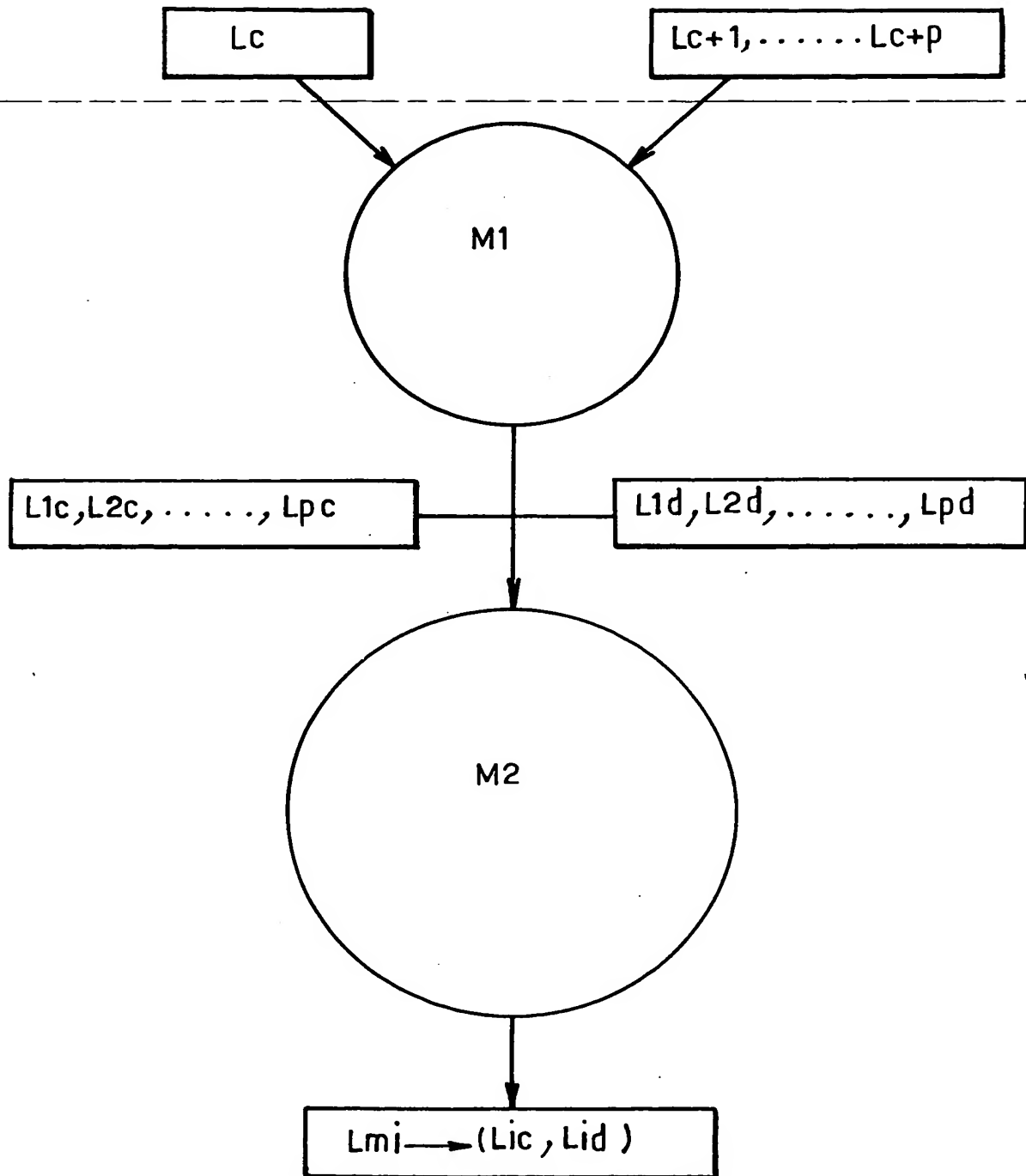


FIG.4b.

FIG.5a.



REVENDEICATIONS

1. Système de développement d'applications interactives distribuées sur un canal de transmission numérique, ces applications interactives installées en un point
- 5 ~~du réseau de transmission numérique consistant en une suc-~~
cession de scènes thématiques incorporant au moins des images en mode point, des images vidéo, des séquences sonores et des informations sous forme de fichier texte représentés au niveau d'au moins un moniteur d'affichage, le
- 10 caractère interactif de ces applications étant produit par l'intermédiaire d'une fonctionnalisation spécifique de l'interface de commande d'un module de commande du moniteur d'affichage muni d'un système d'exploitation, l'ensemble moniteur d'affichage, module de commande et système
- 15 d'exploitation constituant une plate-forme d'exécution, ce système comportant au moins un éditeur électronique permettant la simulation, d'une part, de l'écran d'affichage dudit moniteur d'affichage, et, d'autre part, de l'interface de commande d'un module de commande associé à ce mo-
- 20 niteur d'affichage, et des moyens générateurs d'objets virtuels, spécifiques à l'application considérée, chaque objet virtuel constitué par des composants choisis parmi au moins des images en mode point, des images vidéo, des séquences sonores et des informations sous forme de fi-
- 25 chiers textes étant susceptible d'être édité, affiché et représenté par ledit éditeur électronique pour former au moins une scène constitutive de ladite application, caractérisé en ce qu'il comporte en outre :
- des moyens de commande de mise en corrélation
- 30 d'une pluralité de couples événement/action, un événement étant défini comme la rupture d'un équilibre d'état dudit

système d'exploitation et une action étant définie comme une initialisation respectivement une modification d'au moins un des paramètres d'affichage et de représentation d'une ou plusieurs scènes constitutives de ladite application :

- des moyens de mémorisation de l'ensemble des couples événement/action, mis en corrélation et constituant ladite application interactive.

2. Système selon la revendication 1, caractérisé en ce que lesdits moyens de commande de mise en corrélation d'une pluralité de couples événement/action, comportent des moyens générateurs d'une série d'instructions présentant une structure de contrôle de type liste de requêtes pour "EVENEMENT" IF condition sur toute variable booléenne $B_i \in [B_0, B_1, \dots, B_j, B_N]$ ensemble de variables booléennes vraie THEN "ACTION" où "EVENEMENT" désigne une variable représentative d'un événement et "ACTION" désigne une variable représentative d'une action conditionnée à l'ensemble des variables booléennes, ladite variable "EVENEMENT" vraie constituant pour chaque couple événement/action une rupture d'équilibre d'état dudit système d'exploitation et ladite action constituant une initialisation ou une modification d'au moins un des paramètres d'affichage et de représentation d'une ou plusieurs scènes constitutives de ladite application.

3. Système selon la revendication 2, caractérisé en ce que l'ensemble des couples événement/action est organisé selon une suite de lignes de macro-instructions comprenant chacune un champ de position d'exécution, un champ événement, un champ action, un champ liste de conditions et un champ variable booléenne d'arrêt.

4. Système selon l'une des revendications 1 à 3, caractérisé en ce que ledit moniteur d'affichage, ledit module de commande et ledit système d'exploitation étant ceux d'un récepteur de télévision, ledit éditeur électronique permet la simulation de l'écran d'affichage de ce récepteur de télévision et de l'interface de commande d'un module de télécommande universel associé à ce récepteur de télévision, à une touche spécifique dudit module de télécommande universel simulé étant associé un événement, provoquant la rupture d'équilibre d'état dudit système d'exploitation.

5. Système selon l'une des revendications 1 à 3, caractérisé en ce que ledit moniteur d'affichage, ledit module de commande et ledit système d'exploitation étant ceux d'une station de travail ou d'un micro-ordinateur, ledit éditeur électronique permet la simulation de l'écran d'affichage de ce micro-ordinateur et de l'interface de commande d'un appareil périphérique d'entrée de ce micro-ordinateur, tel que le clavier, à une touche spécifique dudit appareil périphérique d'entrée étant associé un événement provoquant la rupture d'équilibre d'état dudit système d'exploitation.

6. Système selon l'une des revendications 1 à 5, caractérisé en ce que l'ensemble des couples événement/action est mis en corrélation selon une mise en correspondance bi-univoque logique entre événement et action, selon une application interactive native, indépendamment du format d'accès imposé par le système ou plate-forme d'accès à l'application interactive considérée.

7. Système selon l'une des revendications 1 à 6, caractérisé en ce que ledit éditeur électronique, lesdits

moyens générateurs d'objets virtuels spécifiques à l'application et lesdits moyens de commande de mise en corrélation d'une pluralité de couples événement/action comprennent un module logiciel pilote permettant, d'une
5 part, à partir d'une fenêtre d'affichage d'une représentation dudit module de commande et d'appareils périphériques, de recevoir une variable "EVENEMENT" correspondante, et, d'autre part, à partir d'une fenêtre d'affichage de scènes constitutives de ladite application, de provoquer
10 soit l'initialisation, soit la modification de ce celles-ci.

8. Système selon la revendication 7, caractérisé en ce que celui-ci comporte en outre un outil de création de macro-instructions, ledit outil comportant au moins une
15 page écran de mise en corrélation de couples événement/action comprenant au moins une fenêtre d'affichage d'un ensemble d'événements, une fenêtre d'affichage d'une action déclenchée associée à l'un des événements de cet ensemble d'événements et au moins une fenêtre de paramé-
20 trage de l'action déclenchée, des moyens de pointage et de sélection, ladite mise en corrélation de couples événement/action étant effectuée par sélection d'un événement parmi l'ensemble d'événements et de l'action par l'intermédiaire du moyen de pointage.

25 9. Système selon la revendication 7 ou 8, caractérisé en ce qu'il comporte en outre un outil de création d'objets virtuels, ledit outil de création d'objets virtuels comportant au moins une page écran de création d'objets comprenant au moins une fenêtre d'affichage d'une
30 liste d'objets accessibles, cette liste d'objets étant affichée selon un champ de niveau et un champ de désignation

d'objet, une fenêtre de choix de mode d'affichage/représentation de cet objet, un champ de type d'objet permettant de discriminer la nature de cet objet, un champ relatif à un ensemble de paramètres fonctions du média d'affichage/représentation de cet objet.

10. Système selon l'une des revendications 7 à 9, caractérisé en ce qu'il comporte en outre un outil d'édition des médias, ledit outil d'édition des médias comportant au moins une fenêtre page écran "Média" comprenant au moins une fenêtre d'affichage d'une liste des plates-formes d'exécution, une fenêtre d'affichage d'une liste de médias spécifiques susceptibles d'intervenir dans la composition d'une scène, chaque média étant défini par un nom et un attribut, l'attribut alloué à un média donné permettant de discriminer le caractère interne ou externe du média vis-à-vis de l'application considérée.

11. Système selon l'une des revendications 7 à 10, caractérisé en ce qu'il comporte en outre un outil d'édition de phrases sous forme de fichier texte, chaque phrase comportant un numéro d'identification permettant d'affecter cette phrase soit au niveau de l'application, soit au niveau d'une ou plusieurs scènes constitutives de cette application.

12. Système selon l'une des revendications 7 à 11, caractérisé en ce qu'il comporte en outre un outil de création de scène comportant au moins une fenêtre d'affichage d'une liste de scènes types repérées par une désignation spécifique, une fenêtre de définition de paramètres de la scène choisie, paramètres tels que numéro d'identification de scène, palette de couleurs, choix du fond à l'initialisation de la scène, position de départ du

- une table des données caractéristiques de chaque plate-forme;

- une table des événements relatifs à ladite plate-forme ;

5 - une table des actions d'application relatives à ladite plate-forme ;

- une table des fontes disponibles pour ladite plate-forme..

16. Système selon la revendication 15, caractérisé
10 en ce que ladite table des données d'application relatives à ladite plate-forme est définie pour l'application considérée par l'intermédiaire d'un fichier texte.

17. Système selon l'une des revendications 1 à 16, caractérisé en ce que chaque application est définie par
15 un ensemble de listes comportant au moins une liste des attributs d'application, une liste des objets virtuels communs, une liste des couples événement/action, une liste des fontes, une liste des médias, une liste des variables, une liste des phrases, une liste des palettes de couleurs,
20 une liste des couleurs et une liste des scènes.

18. Système selon la revendication 15, caractérisé en ce que chaque élément de la liste des scènes, relatif à une scène considérée, constitutive de cette application, comporte au moins une liste d'attributs de scène, une
25 liste des objets virtuels constitutifs de la scène, une liste d'options, une liste des phrases de la scène.

19. Système selon l'une des revendications 1 à 6, caractérisé en ce que l'ensemble des structures de données et de macro-instructions constituant une application interactive native, celui-ci comporte en outre un module de
30 traduction de ladite application interactive native en une

application interactive dédiée à un terminal d'accès de type déterminé.

20. Système selon l'une des revendications 1 à 19, caractérisé en ce que, dans le but de développer des applications interactives à diffusion cyclique accessibles sur un canal TV à partir d'un terminal d'accès muni d'un décodeur TV, ledit système comporte au moins :

- un moyen gestionnaire d'une pluralité de scènes successives constitutives de cette application selon un niveau de profondeur de chaque scène vis-à-vis de la scène courante, ledit module gestionnaire permettant d'analyser et de discriminer sur un nombre déterminé de scènes successives voisines de la scène courante les objets virtuels communs et les médias communs associés à ces objets de ces scènes successives des objets virtuels distincts et des médias distincts associés à ces objets de ces scènes successives ;

- des moyens d'actualisation desdits objets virtuels et médias communs et des objets virtuels et médias distincts pour au moins une scène successive constitutive de cette pluralité de scènes successives par rapport à la scène courante, ce qui permet d'optimiser le volume de données de réactualisation et le temps d'acquisition de ces données d'actualisation en fonction de la taille mémoire allouée à chaque type de média et du nombre de niveaux de profondeur d'analyse de chaque scène successive.

THIS PAGE BLANK (USPTO)

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☒ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☒ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

THIS PAGE BLANK (USPTO)